W90221F

VERSION 0.4

MARCH, 2000

Table of Contents

1. OVERVIEW

The W90221 is a highly integrated 32-bit processor for a wide range of embedded applications, such as set-top box, web browser and visual/data communication devices. Fig 1-1 shows a block diagram of the overall system. The W90221 consists of the system support logics as well as an embedded 32-bit PA-RISC processor.

The 32-bit PA-RISC core has 4K bytes of instruction cache memory, 4K bytes of data cache memory, a dual-cycle multiply/accumulate module, and integrated functions for interfacing to numerous system components and external I/O modules. Besides, it's designed with a flexible power management scheme (under software control) and lots of low power circuits to eliminate the chip's power consumption. The W90221 consumes only 375 mA as chip operating at its maximum speed.

The 2-D graphic accelerator is the major mega-functional cell integrated in this chip. This unit provides directly connect to TV, analog LCD monitor and CRT monitors, intending for low cost web browser solution. The chip contains an ISA-like bus interface (shared with PCI bus pins) to connect low speed devices, such as code/data ROM/Flash and traditional ISA-like or IDE devices, an EDO/SDRAM controller compliant with PC-100 standard, a PCI bridge supporting up to three external PCI masters, a IEEE-1284 compliant parallel port interface (PPI), two RS-232 type universal asynchronous serial port (UART), two timer channels and a flexible synchronous interface (SSI) connecting to an external audio or telephony codec devices. The overall features are listed section 2.

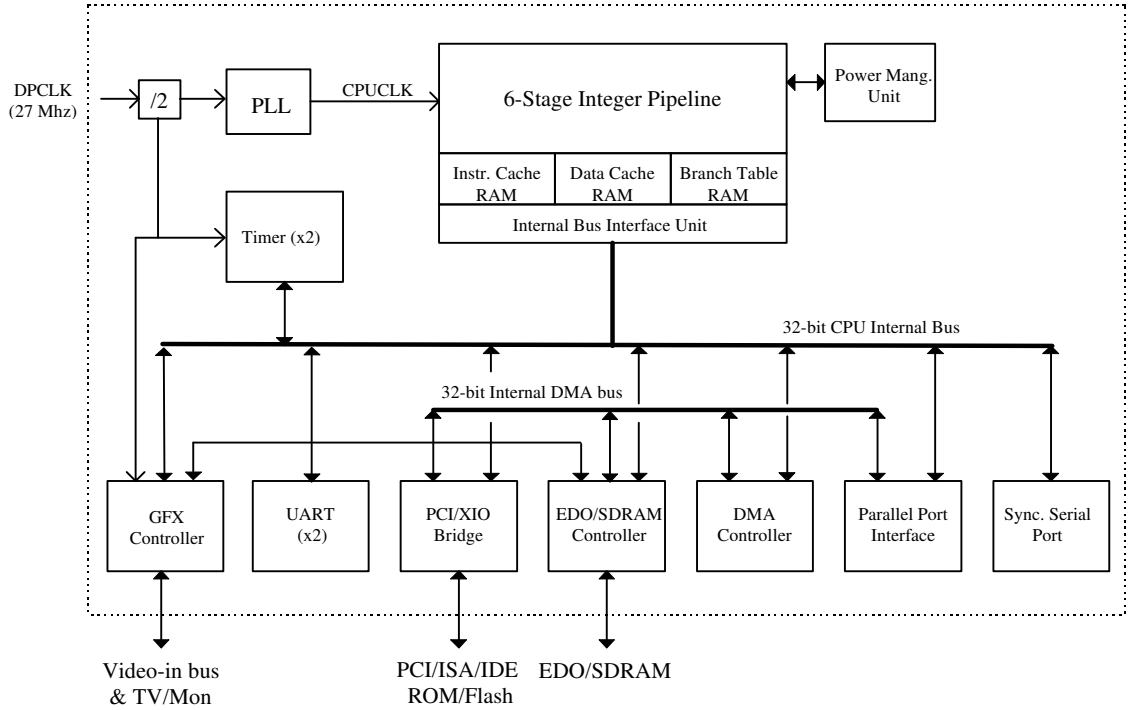


Fig 1-1 : W90221 Internal Block Diagram

2. FEATURES

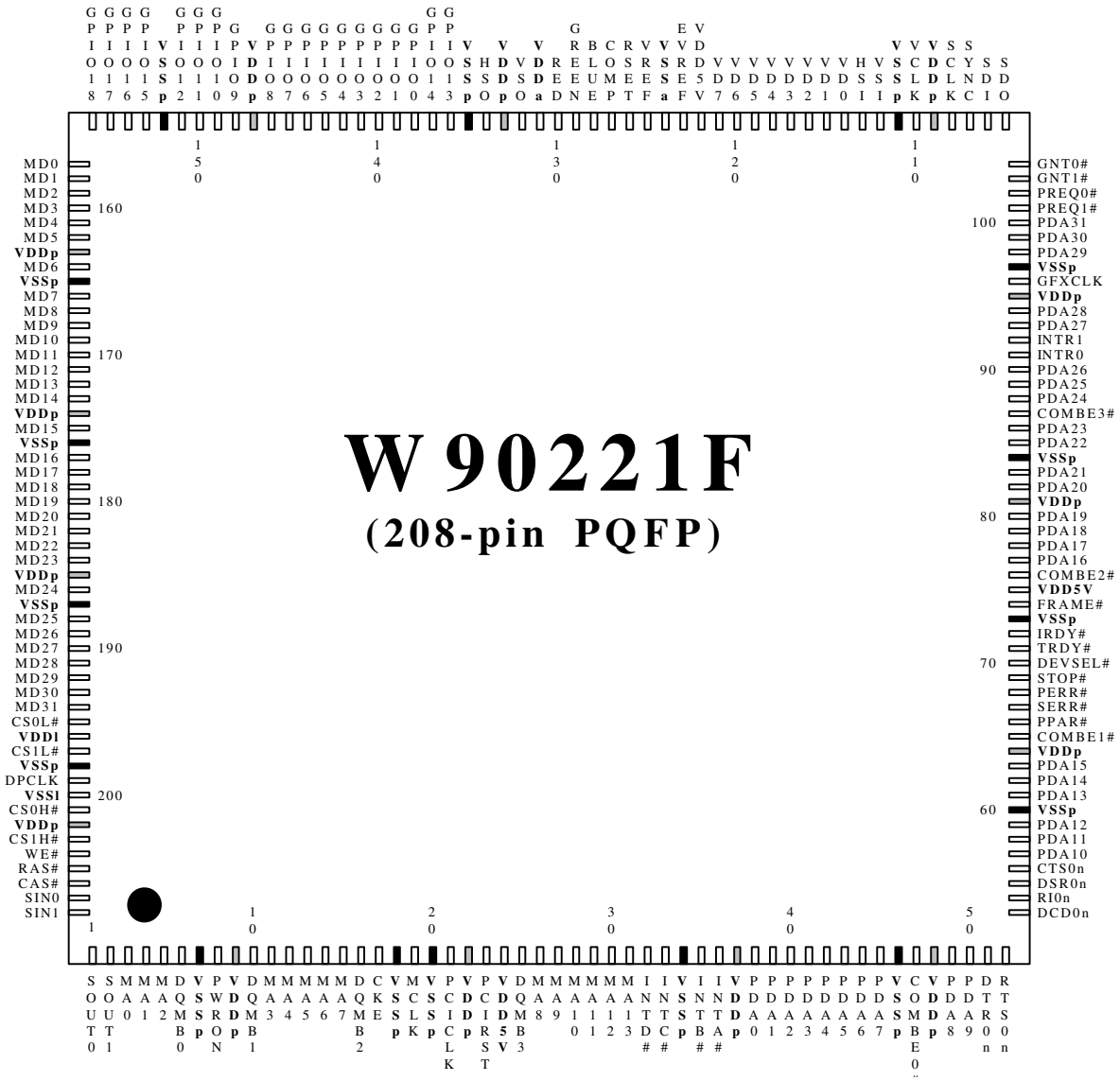
- PQFP 208-pin package

- High level of integration
  - minimal number of inter-chip connections
  - 32-bit PA-RISC core with cache memory, multiply-accumulate module and flexible power management unit
  - a 2-D graphic accelerator directly connect to TV, LCD and monitors
  - a ISA-like bus interface connecting 8-bit ROM/Flash, 8/16 bit ISA or IDE devices
  - SDRAM controller supports EDO type DRAM or PC-100 SDRAM
  - a PCI bridge supports up to three PCI master devices
  - an IEEE-1284 compliant parallel port connecting an external printer
  - two RS-232 compliant serial port connecting external MODEM controller or other serial devices
  - a synchronous serial port connecting external audio or telephony codec devices
  - two timer channels for general purpose usage

- High performance and low power consumption
  - 0.35-micron single-poly-triple-metal CMOS process
  - pure 3.3V logics within SDRAM interface
  - split rail design (3.3V/5V IO and 3.3V core) in other interfaces
  - maximal operation frequency : 150 MHz
  - typical active current : 2.5 mA/MHz
  - typical suspend current (PLL turn off) : -
  - fully static design
  - programmable standby clock to reduce standby current
  - real time clock and UART baud rate base on 13.5 MHz or 18.432 MHz

3. PIN DIAGRAM

The W90221F is available with a 208-pin quad flat pack (PQFP) device configuration, shown below.



4. DETAIL PIN DESCRIPTIONS

The following abbreviations are used for pin types in the following sections : (I) indicates inputs; (O) indicates
outputs; (I/O) indicates a bi-directional signal; (TS) indicates three-state; (OC) indicates open collector. (AO) indicates analog output; (AI) indicates analog input;

PIN Name	DIR	PIN #	DESCRIPTION
System Reset and Clock :			
PWRON	I	8	CPU Power-On reset input, high active
DPCLK	I	199	This clock source serves as internal PLL input as well as VA's system clock. A precise **27MHz** clock source shall be connected to this pin during normal operation.
GFXCLK	I	96	This clock source serves as pixel clock, 36MHz to 50MHz, using in 800x600 non-interlace monitor. For TV subsystem, this clock may pull high or low externally. Meanwhile, GFXCLK may also serves as system OSC (for baud rate or timer adjustment), if **MD[24]** is pull high externally.
General Purpose I/O pins :			
GPIO[0:7]	I/O	138-145	If parallel port is enable (port 0x3e[4] = 1), these pins serve as bi-directional ECP data bus "**ED[0:7]**" with ED[0] is the most significant bit (inout). If parallel port is not enable (port 0x3e[4:5] = 0x), these pins provide general purpose I/O functionality (inout).
GPIO[8]	I/O	146	If parallel port is enable (port 0x3e[4:5] = 1x), this pin serve as ECP "**nInit**" (output). If parallel port is not enable (port 0x3e[4:5] = 0x), this pin provides general purpose I/O functionality (inout)
GPIO[9]	I/O	148	If parallel port is enable (port 0x3e[4:5] = 1x), this pin serve as ECP "**nSelectIn**" (output). If parallel port is not enable (port 0x3e[4:5] = 0x), this pin provides general purpose I/O functionality (inout)
GPIO[10]	I/O	149	If parallel port is enable (port 0x3e[4] = 1x), this pin serve as ECP "**Select**" (input). If parallel port is not enable (port 0x3e[4:5] = 00), this pin provides general purpose I/O functionality (inout). During "clock test" mode (port 0x3e[4] = 01), this pin outputs internal **CPUCLK** (output).
GPIO[11]	I/O	150	If parallel port is enable (port 0x3e[4] = 1x), this pin serve as ECP "**PError**" (input). If parallel port is not enable (port 0x3e[4:5] = 00), this pin provides general purpose I/O functionality (inout). During "clock test" mode (port 0x3e[4] = 01), this pin outputs internal **MCLK_ctl** (output).

GPIO[12]	I/O	151	If parallel port is enable (port 0x3e[4] = 1x), this pin serve as ECP "**nFault**" (input). If parallel port is not enable (port 0x3e[4:5] = 00), this pin provides general purpose I/O functionality (inout). During "clock test" mode (port 0x3e[4] = 01), this pin outputs internal **MCLK_data** (output).
GPIO[13:14]	I/O	137, 138	These two pins always provide general purpose I/O functionality.
GNT2#/ nAutoFd	O	153	If parallel port is enable (port 0x3e[4] = 1x), this pin serve as ECP "**nAutoFd**". If parallel port is not enable (port 0x3e[4:5] = 0x), this pin outputs PCI bridge Grant two "GNT2#".
GNT3#/ nStrobe	O	154	If parallel port is enable (port 0x3e[4] = 1x), this pin serve as ECP "**nStrobe**". If parallel port is not enable (port 0x3e[4:5] = 0x), this pin outputs PCI bridge Grant three "GNT3#".
PREQ2#/ nAck	I	155	If parallel port is enable (port 0x3e[4] = 1x), this pin serve as ECP "**nAck**". If parallel port is not enable (port 0x3e[4:5] = 0x), this pin inputs Master request two "PREQ2#".
PREQ3#/ Busy	I	156	If parallel port is enable (port 0x3e[4] = 1x), this pin serve as ECP "**Busy**". If parallel port is not enable (port 0x3e[4:5] = 0x), this pin inputs Master request three "PREQ3#".
PCI/AIO Bus Bridge :			for more detail description of the PCI signals please refer to the **PCI LOCAL BUS SPECIFICATION**
INTD#/ **XGLBCS#**	O	32	During **PCI** cycles : If AIO is enable, this signal shall not connect to any PCI bus master During **AIO** cycles : Asserted low indicating a AIO command cycle is ongoing
PDA[31:24]/ **XA[8:15]/** **XD[15:8]**	I/O	100-98, 94, 93, 90-88	During **PCI** cycles : These pins serve as highest byte of PCI 32-bit address/data bus. During **AIO memory** cycles0 : These pins serve as highest byte of 24-bit address lines (XA[8:31]) During **AIO IO** cycles. : These serve as high byte of 16-bit data lines (XD[15:0]).
PDA[23:8]/ **XA[16:31]**	I/O	85, 84, 83, 82, 80-77, 63-61, 59-57, 50, 49	During **PCI** cycles : These pins serve as bits 16-31 of PCI 32-bit address/data bus. During **AIO** cycles : These pins serve as lower 16-bit of 24-bit address lines (XA[8:31]).
PDA[7:0]/ **XD[15:8]**	I/O	45-38	During **PCI** cycles : These pins serve as lowest byte of PCI 32-bit address/data bus. During **AIO memory** cycles : These pins serve as the 8-bit data lines. During **AIO IO** cycles : These pins serve as low byte of 16-bit data lines (XD[15:0]).
COMBE[3]/ **AIOCS#**	I/O	87	During **PCI** cycles : Bit-3 of command/byte bus During **AIO** cycles : AIO chip-select for its IO devices
COMBE[2]/ **XROMCS#**	I/O	76	During **PCI** cycles : Bit-2 of command/byte bus During **AIO** cycles : AIO chip-select for its memory devices

COMBE[1]/ **XWR#**	I/O	65	During **PCI** cycles : Bit-1 of command/byte bus During **AIO** cycles : Asserted low, if INTD# is also low, indicating a AIO write command cycle is ongoing
COMBE[0]/ **XRD#**	I/O	47	During **PCI** cycles : Bit-0 of command/byte bus During **AIO** cycles : Asserted low, if INTD# is also low, indicating a AIO read command cycle is ongoing
INTA#, INTB#, INTC#	I	36, 35, 33	PCI Interrupt input, level sensitive, low active signal. Once the INTx# signal is asserted, it remains asserted until the device driver clear the pending request. When the request is cleared, the device de-asserts its INTx# signal.
PREQ0#, PREQ1#	I	102, 101	PCI Request input, indicates to the PCI arbiter that this agent desires use of the bus.
GNT0# GNT1#	O	104, 103	PCI Grant output, indicates to the agent that access to the bus has been granted.
PCIRST#	O	23	PCI Reset output, is used to bring PCI-specific registers, sequencers, and signals to a consistent state. Low active.
PCICLK	O	21	PCI Clock output, provides timing for all transactions on PCI and is an input to every PCI device.
SERR#	I	67	PCI System Error is for reporting address parity errors, data parity errors on the Special Cycle command, or any other system error where the result will be catastrophic. The assertion of SERR# is synchronous to the clock and meets the setup and hold times of all bused signals.
PERR#	I/O	68	PCI Parity Error is only for the reporting of data parity errors during all PCI transactions except a Special Cycle. The PERR# pin is sustained tri-state and must be driven active by the agent receiving data two clocks following the data when a data parity error is detected. The minimum duration of PERR# is one clock for each data phase that a data parity error is detected. An agent cannot report a PERR# until it has claimed the access by asserting DEVSEL# (for a target) and completed a data phase or is the master of the current transaction.
STOP#	I/O	69	PCI Stop indicates the current target is requesting the master to stop the current transaction.
TRDY#	I/O	71	PCI Target Ready indicates the selected device ability to complete the current data phase of the transaction. A data phase is completed on any clock both TRDY# and IRDY# are sampled asserted. During a read, TRDY# indicates that valid data is present on PDA[31:0]. During a write, it indicates the target is prepared to accept data. Wait cycles are inserted until both IRDY# and TRDY# are asserted together.
DEVSEL#	I/O	70	PCI Device Select, when actively driven, indicates the driving device has decoded its address as the target of the current access. As an input, DEVSEL# indicates whether any device on the bus has been selected.

FRAME#	I/O	74	PCI Cycle Frame is driven by the current master to indicate the beginning and duration of an access. FRAME# is asserted to indicate a bus transaction is beginning. While FRAME# is asserted, data transfers continue. When FRAM# is deasserted, the transaction is in the final data phase or has completed.
IRDY#	I/O	72	PCI Initiator Ready indicates the bus master ability to complete the current data phase of the transaction. A data phase is completed on any clock both IRDY# and TRDY# are sampled asserted. During a write, IRDY# indicates that valid data is present on PDA[31:0]. During a read, it indicates the master is prepared to accept data. Wait cycles are inserted until both IRDY# and TRDY# are asserted together.
PPAR	I/O	76	PCI Parity is even parity across PDA[31:0] and C/BE[3:0]#. PPAR is stable and valid one clock after the address phase. For data phases, PPAR is stable and valid one clock after either IRDY# is asserted on a write transaction or TRDY# is asserted on a read transaction. (PPAR has the same timing as PDA[31:0], but it is delayed by one clock.) The mater drives PPAR for address and write data phases; the target drives PPAR for read data phase.
8-bit Video-In (VMI) Bus :			
VIN[0:7]	I/O	114~121	During **normal mode**, this bus **inputs** 8-bit digital components of YCbCr 4:2:2 video-in data from external video controller (ex. TV decoder or MPEG decoder). During **test mode**, this bus **outputs** 8-bit digital components of YCbCr 4:2:2 video-out generated by internal video accelerator (VA).
HSI	I	113	Horizontal Sync of video-in frames. The content of VAconf[?] determines the polarity of this signal.
VSI	I	112	Vertical Sync of video-in frames. The content of VAconf[?] determines the polarity of this signal.
VCLK	I	110	This clock source serves as VMI bus pixel clock (27MHz). A precise **27MHz** clock source shall be connected to this pin.
Display and DAC interface :			
CP2/C/Blue	AO	128	During **composite** video mode (NTSC, PAL), this pin is analog "composite video" output. During **S-Video** mode, this pin is analog "**chrominance**" output. During **monitor** mode, this analog output supplies current corresponding to the "**blue**" intensity of the pixel being displayed. (To maintain IBM VGA compatibility, R-G-B outputs are typically terminated to monitor's ground with a 75 omv 2% resistor. This resistor, in parallel with the 75 omv resistor in the monitor, will yield a 37.5 omv impedance to ground. For a full-scale voltage of 700 mV, full-scale current output will be **18.7mA**.)

CP1/Y/Green	AO	129	During **composite** video mode (NTSC, PAL), this pin is analog "composite video" output. During **S-Video** mode, this pin is analog "**luminance**" output. During **monitor** mode, this analog output supplies current corresponding to the "**green**" intensity of the pixel being displayed.
CP0/Red	AO	130	During **composite** video mode (NTSC, PAL), this pin is analog "composite video" output. During **S-Video** mode, this pin left no connection. During **monitor** mode, this analog output supplies current corresponding to the "**red**" intensity of the pixel being displayed.
HSO	O	134	Horizontal Sync of the displayed graphic output. The content of VAconf[?] determines the polarity of this signal.
VSO	O	132	Vertical Sync of the displayed graphic output. The content of VAconf[?] determines the polarity of this signal.
VREF	AO	125	Voltage Reference Out; Bypass and decouple the voltage reference with 0.1uF ceramic capacitor to the TVDD. The decoupling capacitor shall be as close to the chip as possible. This pin as well as "COMP" are used to control the current of internal current sources are exactly equal to "Iref".
VREF	AO	125	Voltage Reference Out; Bypass and decouple the voltage reference with 0.1uF ceramic capacitor to the TVDD. The decoupling capacitor shall be as close to the chip as possible. This pin as well as "COMP" are used to control the current of internal current sources are exactly equal to "Iref".
COMP	AO	127	Compensation pin. It shall be decoupled with a 0.1uF ceramic capacitor to TVDD. The decoupling capacitor shall be as close to the chip as possible.
RSET	AO	126	Current Source Adjusting Resistor. This pin is used to adjust the full scale current of TV's analog outputs. A resistor shall be connected between this pin and TVSS. (The DAC's "Iref" of current mirrors are adjusted by this pin). The **Iref** is approximate to **1.16V/RSET**.
EXTVREF	AI	123	External Vref input. This signal supplies the DAC's "bandgap" output from a external 1.235V voltage source. A 0.1uF bpass capacitor should be always connected between this pin and TVDD. ("bandgap" is an voltage stabilizer of voltage-reference-generator "Vref"). This pin may left unconnected.
Memory Controller Interface :			
CS0L#/RAS0#, CS1L#/RAS1#	O	195, 197	During **EDO mode**, these signals are served as RAS0#, RAS1# that used to latch the row address MA[0:11] lines into the DRAM. Each signal is used to select one DRAM bank. During **SDRAM mode**, these signals are served as CS0L#, CS1L# that indicates the command decoder is enable or disable.

CS0H#, CS1H#	O	201, 203	Similar to CS0L#/CS1L#, when on-board SDRAM is used, These pins are NC pins. When SDRAM DIMM module is used, these signals indicates current cycle accessing the high word (32 bits) of DIMM's data bus (double words).
RAS#, CAS#	O	205, 206	This signals along with WE# and CS# define the command code of SDRAM configuration cycles.
WE#	O	204	This signal asserted to indicate a write cycle to DRAM.
CAS[0:3]#/ DQMB[0:3]	O	6, 10, 16, 25	In **EDO mode**: these signals are served as CAS# function and used to latch the column address (MA[0:11]) into DRAMs. it also indicates which bytes can be accessed. In **SDRAM mode** : these signals are served as DQMB function, these are input mask signals for write cycle and output enable signals for read cycle.
MA[0:13]	O	3, 4, 5, 11, 12, 13, 14, 15, 26, 27, 28, 29, 30, 31	These signals are used to provide the multiplexed row and column address to the EDO DRAM or SDRAM.
MD[0:31]	I/O	157-162, 164, 166-173, 175, 177-184, 186, 188-194	These signals are used to interface to the DRAM data bus.
CKE	O	17	This signal are used to enable or disable MCLK into SDRAM.
MCLK	O	19	This signal is SDRAM clock input, all SDRAM input /ouput signals are refrenced with MCLK rising edge.
COM0 Serial Port Signal :			
SIN0	I	207	COM0 serial data input from the communication link (modem or peripheral device).
SOUT0	O	1	COM0 serial data output to the communication link (modem or peripheral device).
CTS0n	I	56	COM0 clear to send signal
DSR0n	I	55	COM0 data set ready
DTR0n	O	51	COM0 data terminal ready
RTS0n	O	52	COM0 request to send
DCD0n	I	53	COM0 data carrier detect
RI0n	I	54	COM0 ring indicator
COM1 Serial Port Signal :			
SIN1	I	208	COM1 serial data input from the communication link (modem or peripheral device).
SOUT1	O	2	COM1 serial data output to the communication link (modem or peripheral device).
Synchronous Serial Port Signal :			
SDI	I	106	Serial data-in from a external codec device
SDO	O	105	Serial data-out to a external codec device

SYNC	I/O	107	Frame sync of SDI/SDO. This signal is an input signal during "slave mode" (CFGH[2] set low)or output signal during "master mode" (CFGH[2] set high)
SCLK	I/O	108	Serial Clock for SDI/SDO transferring. This signal is an input signal if **MD[25]** is pulled **low**, and act as output signal during is MD[25] is pull **high**.
Miscellaneous :			
INTR0	I	91	This pin serves as an external interrupt request. A active high-state in this pin will make EIER[12] be set. This pin may also serve as an interrupt request pin for an IDE slot.
INTR1	I	92	This pin serves as an external interrupt request. A active high-state in this pin will make EIER[13] be set. This pin may also serve as an interrupt request pin for an IDE slot.
Power/Ground pin :			
VDD5V	I	24,75,122	5.0V Vdd (for a mixed 5.0V/3.3V environment)
VDDp	I	9,22,37,48,64, 81,95,109,133 147,163, 174, 185,202	Global 3.3V Vdd
VSSp	I	7,18,20,34,46, 60,73,84,95, 111,135,152, 165,176,187, 198,	Global VSS
VDDa	I	131	3.3V Vdd for DAC
VSSa	I	124	VSS for DAC
VDDl	I	196	3.3V Vdd for PLL
VSSl	I	200	VSS for PLL

=

5 MEGACELLS

5.1 CLOCK MODULE AND PLL

Overview :
(Left for Blank)

Block Diagram :

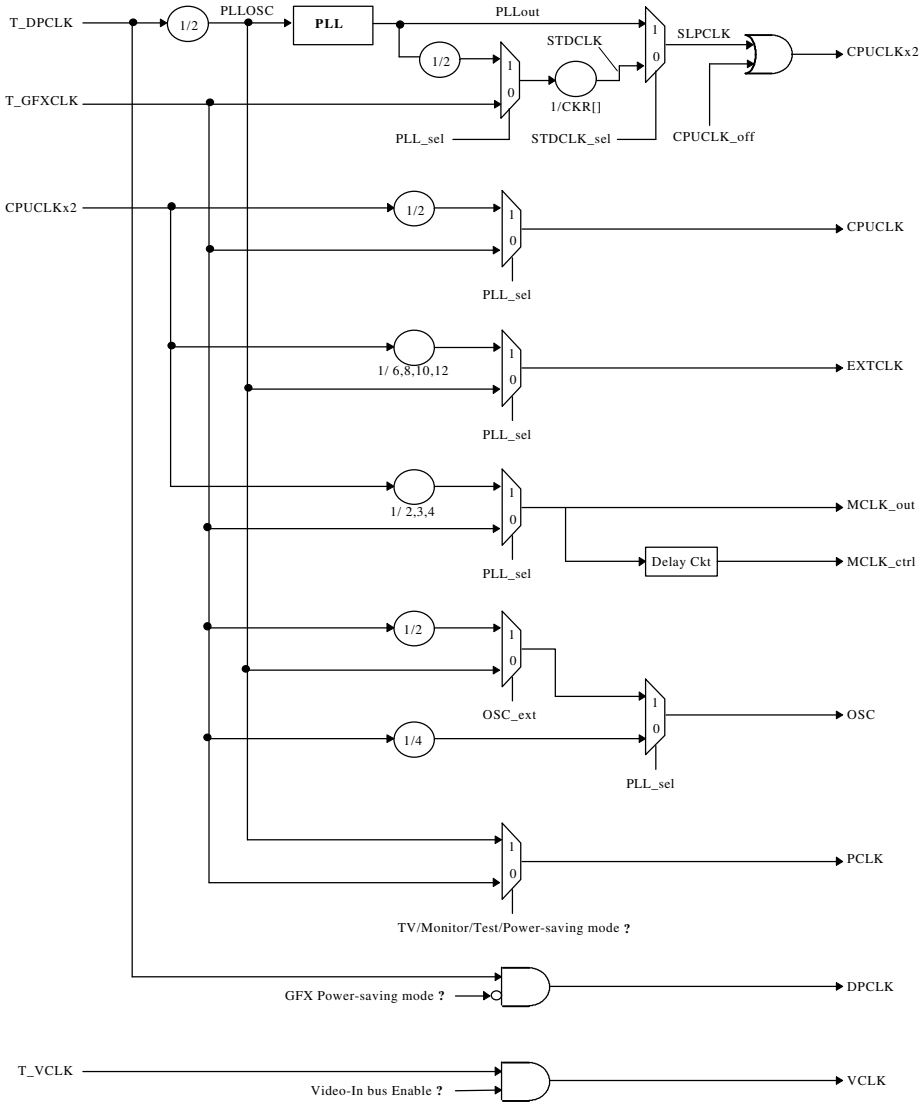


Fig 5.1.1 CLock-Generator :

Features :

- Single **27MHz** oscillator to generate all clock sources
  - PLL input clock (13.5MHz)
  - CPU clock range : 80, 90, 100, 110, 120, 130, 150, 180MHz
  - Video-Accelerator clock (27MHz, 13.5MHz)
  - system clock (CPU-clock/2, /3, /4)
  - SDRAM clock (CPU-clock/1, /1.5, /2)
  - UART, Timer clock (13.5MHz)
- Optional SCLK input for SSI interface as SSI operation in slave mode

Power-on setting :

- MD[**31**] pull high to enable internal PLL unit
  MD[**28-30**] used to choose CPUCLK from 80MHz to 180MHz,
  MD[**26-27**] set 2'b00 to choose 1/3 CPUCLK,
  2'b01 to choose 1/4 CPUCLK,
  2'b10 to choose 1/5 CPUCLK
  2'b10 to choose 1/6 CPUCLK as EXTCLK
  MD[**25**] pull high to set X_SCLK (of SSI unit) to output mode (SCLK master mode)
  MD[**24**] set high to choose OSC clock as GFXCLK/2
  MD[**20-23**] are reserved for further extension

- CTM[0-3] (0xf00001d8) memorize states of **MD[20~23]** during power-on interval (refer to pp.66)
  (Firmware reads these bits to know what kind of target board is operating.)
- DRAMctrl2[2-3] (0xf000003e) defines MCLK frequencies (refer to pp.28)
- TVTWH[12-15] (0xf0000178) defines the format of video output (refer to pp.52)
- VPTC[17] (0xf000017c) defines the direction of 8-bit video-in bus (refer to pp.53)

Related Pins :

- DPCLK (input) :
  This clock source serves as internal PLL input as well as VA's system clock. A precise 27MHz clock source
  shall be connected to this pin during normal operation.

- VCLK (input) :
  This clock source serves as VMI bus pixel clock (27MHz). A precise 27MHz clock source shall be connected to
  this pin.

- GFXCLK (input) :
  This clock source serves as pixel clock, 36MHz to 50MHz, using in 800x600 non-interlace monitor. For TV
  system, this clock may pull high or low externally. Meanwhile, GFXCLK may also serves as system OSC (for baud
  rate or timer adjustment), if MD[24] is pull high externally.

- SCLK (in/out) :
  This pin is the serial bit clock between SSI and codec devices. The SCLK may be input or output depending
  on SSI operated in slave- or master-mode respectively.

- VDDL, VSSL :
  Dedicate power/ground pins for internal PLL unit. VDDL shall connects a 3.3V voltage source.

Operation Modes :

- Normal Mode (PLL is enable) :

A. TV system : DPCLK = 27MHz
VCLK = 27MHz
GFXCLK may be unconnected

PCLK = DPCLK/2 = 13.5MHz
PLLin = DPCLK/2 = 13.5MHz
CPUCLK = 80 ~ 180MHz
EXTCLK = CPUCLK/3, /4, /5, /6
MCLK = CPUCLK/1, /1.5, /2
OSC = **DPCLK/2 = 13.5MHz** (MD[24] = 0 on power-on reset)
= **GFXCLK/2 = 18.432MHz** (MD[24] = 1 on power-on reset)

B. Monitor : DPCLK = 27MHz
VCLK = 27MHz
GFXCLK = **36.864MHz** (for derivating 18.432MHz OSC)

PCLK = GFXCLK = 36.864MHz
PLLin = DPCLK/2 = 13.5MHz
CPUCLK = 80 ~ 180MHz
EXTCLK = CPUCLK/3, /4, /5, /6
MCLK = CPUCLK/1, /1.5, /2
OSC = **DPCLK/2 = 13.5MHz** (MD[24] = 0 on power-on reset)
= **GFXCLK/2 = 18.432MHz** (MD[24] = 1 on power-on reset)

- Test Mode (PLL is disable) : TV system always

DPCLK = 27MHz
VCLK = 27MHz
GFXCLK = 66MHz
PCLK = DPCLK/2 = 13.5MHz
CPUCLK = GFXCLK = 66MHz (for instance)
EXTCLK = CPUCLK/2 = 33MHz
MCLK = CPUCLK = 66MHz
OSC = CPUCLK/4 = 16.5MHz

5.2 PA-RISC CPU CORE

Overview :
(Left for Blank)

Block Diagram :
(Left for Blank)

Features :

- Base on PA-RISC 1.1 level-0 architecture
- 32-bit integer instruction set and register files
- Maximum 100 MHz operation frequency

- 3.3V and 0.01W/MHz at full speed operation
- On-chip power management
  - Build-in software-independent `dynamic power-down mode`
  - Programmable `stand-by ` and `sleep ` mode
  - Specific instruction to assist power-down control and ICE function
- High-speed 32-bit integer pipeline design
  - 6 stages for Load/Store instructions
  - 5 stages for other instructions
- On-chip cache memory
  - 4 KB, direct-map instruction cache and 4 KB, 4-way set-associative data cache
  - Write-through and write-back support for data cache
  - One level read buffer and wrap-around support in each cache
  - One level write buffer and hit-under-miss support in data cache
  - Cache-locking support in instruction cache
- Dynamic branch prediction
  - Build-in 1-level 256 entry, 4-way set-associative (LRU) Branch-Target-Buffer to improve branch prediction rate and accelerate pipeline throughput
- One high speed (2 CPU cycles) 16-/32-bit MAC and multimedia extended instructions have been built-in for DSP related calculation
- Specific serial-ICE-interface to facilitate chip debugging and software development

Related Pins :

- PWRON (input) :
  System Power-On Reset signal; Set this signal to logic high will reset the chip and force all megacells returned to their initial states.

- INTR0 (input) :
  This pin serves as an "external interrupt request". Set this signal to logic 1 will also set EIER[12] to logic 1.
  (The 16-bit IDE slot can use this pin as its interrupt request).

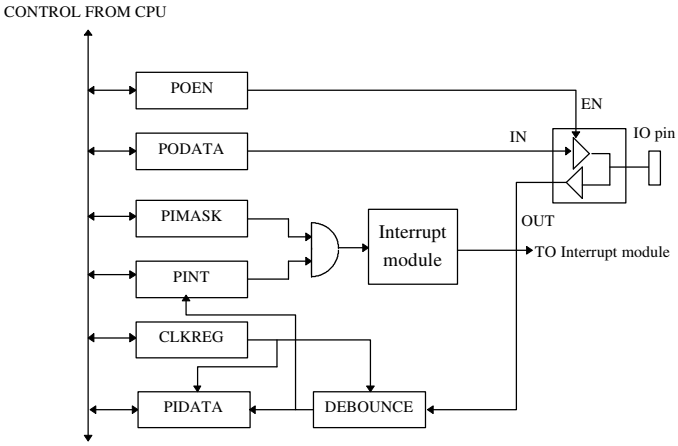
- INTR1 (input) :
  This pin serves as another "external interrupt request". Set this signal to logic 1 will also set EIER[13] to logic 1.
  (The other 16-bit IDE slot can also use this pin as its interrupt request).

Operation Modes :
(Left for Blank)

5.3 GPIO

Overview :
The W90221 provides totally 19 gpio pins. These pins may serves as traditional PIO, or parallel port interface, or another 2 PCI bus master request/grant, depend on what **bit[4:5]** of port **0xf000003e** are set. Right after power-on reset,
all these pins are hi-Z (input mode).

Block Diagram :



Features :

- Each PIO port can generate a separate positive and negative edge interrupt.
- Each PIO port consists of a bi-directional buffer connected to the appropriate W90221 pin
- The input buffer is routed directly to a debounce circuit.
- The debounce circuit performs a 2 TCLK_BUN clock debounce of the input signal.
- Programmable debounce circuit sampling clocks(TCLK_BUN), the clock range is TCLK ~ TCLK/128.

Related Pins :

Table 5.3-1 : GPIO definitions

# of GPIO	cfg = 00	cfg = 01	cfg = 1x
GPIO[0:7]	PIO[0:7] (io)	PIO[0:7] (io)	ED[0:7] (io)
GPIO[8]	PIO[8] (io)	PIO[8] (io)	nInit (out)
GPIO[9]	PIO[9] (io)	PIO[9] (io)	nSelectIn (out)
GPIO[10]	PIO[10] (io)	**CPUCLK** (out)	Select (in)
GPIO[11]	PIO[11] (io)	**MCLK_ctl** (out)	PError (in)
GPIO[12]	PIO[12] (io)	**MCLK_data** (out)	nFault (in)
GPIO[13]	PIO[13] (io)	PIO[13] (io)	PIO[13] (io)
GPIO[14]	PIO[14] (io)	PIO[14] (io)	PIO[14] (io)
GPIO[15]	**GNT2# (out)**	**GNT2# (out)**	nAutoFd **(out)**
GPIO[16]	**GNT3# (out)**	**GNT3# (out)**	nStrobe **(out)**
GPIO[17]	**PREQ2# (in)**	**PREQ2# (in)**	nAck **(in)**
GPIO[18]	**PREQ3# (in)**	**PREQ3# (in)**	Busy **(in)**

Note : cfg = bit[4:5] of port 0xf000003e

Operation Modes :

The following figure showes pio timing diagram for the input pin data -- the shortest time, the pio interrupt register will generate an interrupt and will latch the pin input data into the input data register.

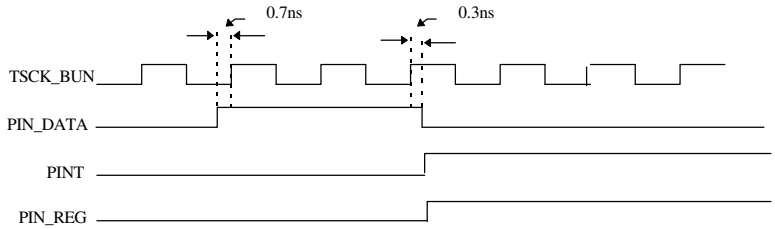


Fig5.3.1 Shortest time pin_data should keep to generate interrupt and latch the input data to pio

The following figure shows PIO timing diagram for the input pin data -- the longest time, the PIO interrupt register will not generate an interrupt and will not latch the pin input data into the input data register.

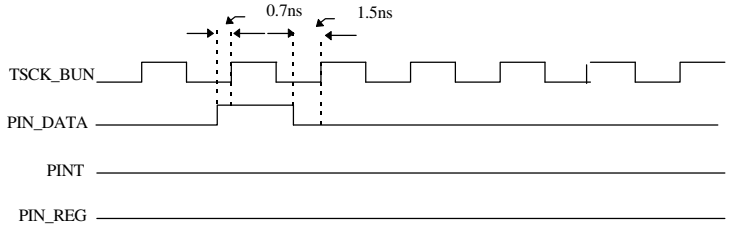


Fig5.3.1 Pin_data keep the longest time will not generate interrupt and will not latch the input data to pio

5.4 MEMORY CONTROLLER

Overview :
The MEMC module within W90221 contains configuration register, control register, timing control registers and other logic to provide 32 bits SDRAM/EDO interface with external SDRAM/EDO memory device, the flexible timing programming can achieve you use different speed of SDRAM/EDO whatever it is 32 bits on board or 64 bits DIMM module.

Block Diagram :

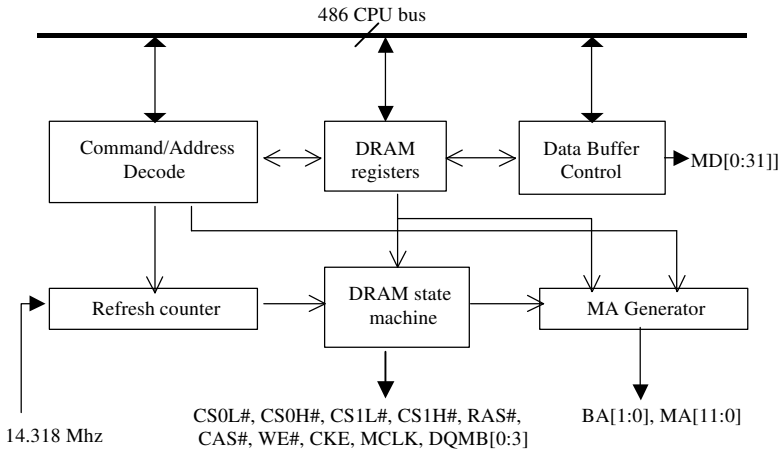


Fig 5.4-1 MEMC Block Diagram

Features :

- supports up to 2 banks of EDO-DRAM (SIMM) or SDRAM (DIMM or ON BOARD)
- 32-bit data interface
- CAS#-befor-RAS# refresh cycles for DRAM module
- programmable RAS#/CAS# timing for DRAM access
- programmable address setup time for DRAM access.
- only supports a burst length of one and burst type of sequential.
- programmable CAS# latency access time.
- provide 1M, 2M, 4M, 8M, 16M DRAM with page size 256 bytes, 512 bytes, 1K bytes, 2K bytes, 4K bytes configuration.

Related Pins :

- CS0L#/RAS0#, CS1L#/RAS1#_ (out) :
  During EDO mode, these signals are served as RAS0#, RAS1# that used to latch the row address on the MA[0:11] lines into the DRAM. Each signal is used to select one DRAM bank.
  During SDRAM mode, these signals are served as CS0L#, CS1L# that indicates the command decoder is enable or disable.

- CS0H#, CS1H# (out) :
  Same as above CS0L#/CS1L#, when using on board SDRAM, it has been reserved and no effect on access, when using external DIMM module, these signals indicates current access on the high 32 bits of 64-bit DIMM.

- RAS#, CAS# (out) :
  This signals along with WE# and CS# define the command is being entered when using SDRAM configuration.

- WE# (out) :
  This signals asserted indicates a write cycle to DRAM.

- CAS[0:3]#/DQMB[0:3] (out) :
  EDO mode: these signals are served as CAS# function and used to latch the column address on the

MA[0:11] lines into th DRAMs. it also indicates which bytes can be accessed.
SDRAM mode : these signals are served as DQMB function, these are input mask signals for write cycle and output enable signals for read cycle.

- MA[0:13] (out) :
These signals are used to provide the multiplexed row and column address to the EDO DRAM or SDRAM.

- MD[0:31] (in/out) :
These signals are used to interface to the DRAM data bus.

- CKE (out) :
This signal are used to enable or disable MCLK into SDRAM.

- MCLK(out) :
This signal is SDRAM clock input, all SDRAM input /ouput signals are refrenced with MCLK rising edge.

Operation Modes :

- MX1 Mode : Once DRAMTctrl2[2:3] is set to 00, memory controller frequency is same as CPUCLK.
- MX1.5 Mode : Once DRAMTctrl2[2:3] is set to 01, Memory controller frequency is CPUCLK/1.5.
- MX2 Mode : Once DRAMTctrl2[2:3] is set to 10, Memory controller frequency is CPUCLK/2.
- MCLK skew control : The SDRAM's CLK and internal MEMC system clocks are adjustable for SDRAM operating in higher clock rate (larger than 80 MHz). Three bit groups are used to define these clocks' skew, 1 SDRAM CLK and two internal MEMC system clock. Following are some suggested setting as SDRAM operated in different modes : (Refer to DRAMctrl definition for details)

A. for MX1 :
DRAMctrl[9:11] = 110,
DRAMctrl[12:14] = 010,
DRAMTctrl[9:11] = 001.

B. for MX1.5 :
DRAMctrl[9:11] = 010,
DRAMctrl[12:14] = 010,
DRAMTctrl[9:11] = 010.

C. for MX2 :
DRAMctrl[9:11] = 110,
DRAMctrl[12:14] = 001,
DRAMTctrl[9:11] = 110.

Application Notes :
The MEMC supports both SDRAM and EDO-RAM, while only 32-bit data bus is available. When 64 bit

SDRAM DIMM is used, special data bus routing is needed (for detail, refer to "Intel PC SDRAM Unbuffered DIMM specification"). The following figures show typical connections and timing between W90221 and SDRAM :

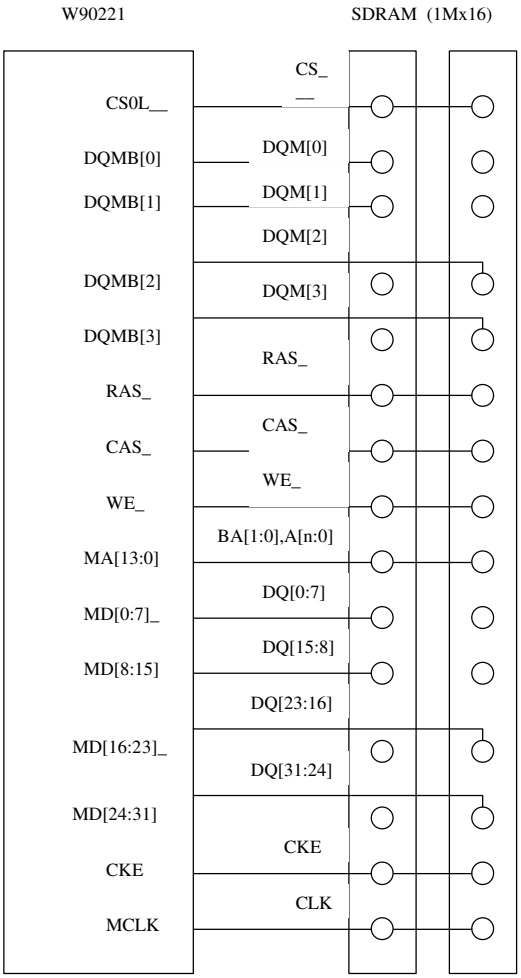


Fig 5.4-2 ON BOARD SDRAM CONNECTION

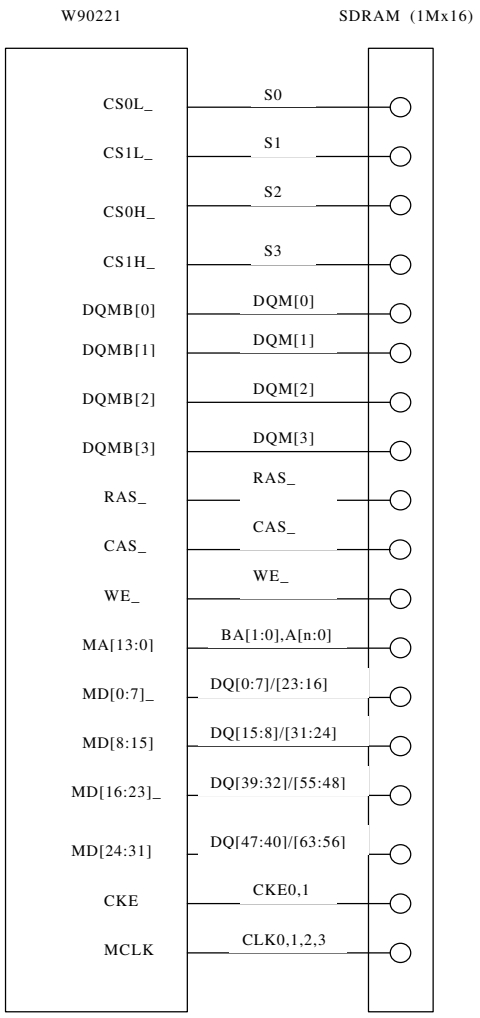


Fig 5.4-2 EXTERNAL DIMM CONNECTION

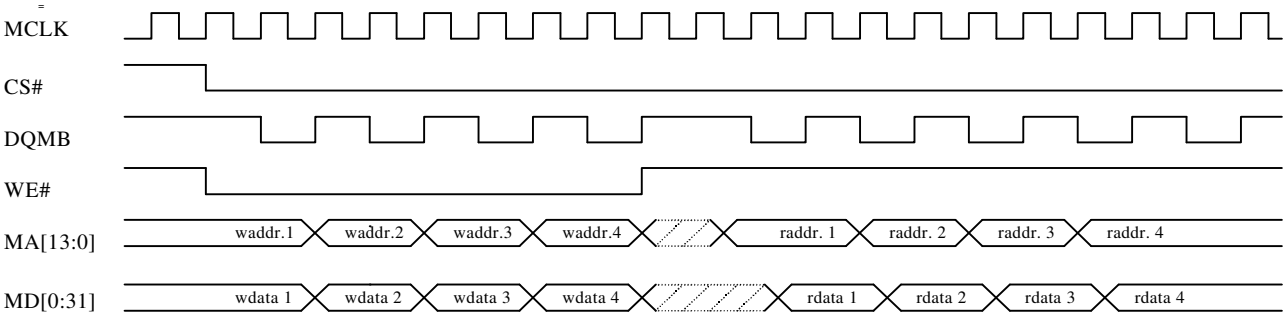


Fig 5.4.3 Fastest EDO memory write/read cycle (32 bits)

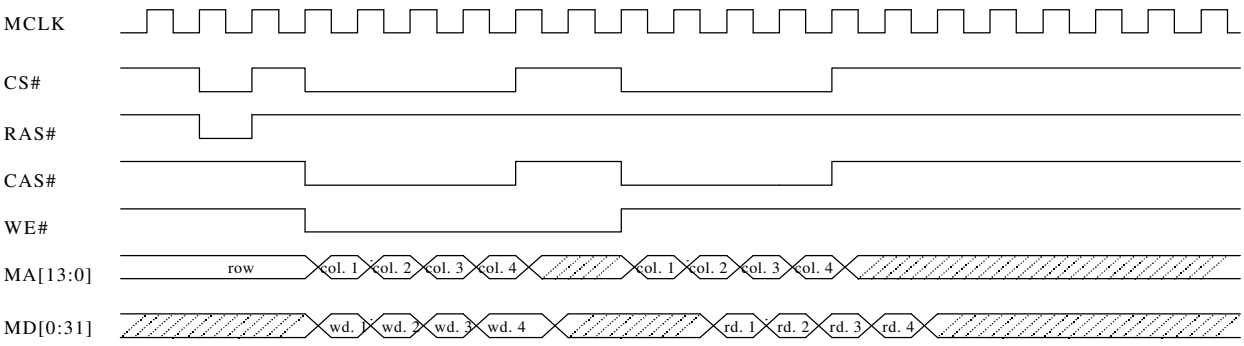


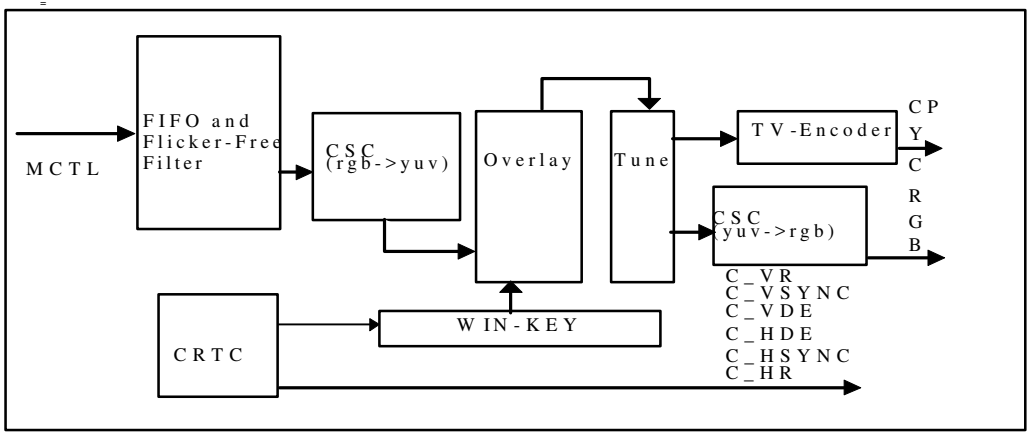
Fig 5.4.4 Fastest SDRAM memory write/read cycle (32 bits)

5.5 VIDEO ACCELERATOR (VA)

Overview :

Display Controller controls the display timing and data to meet the requirements of display devices. The display controller of W90221 supports RGB monitor output and S-Video, RCA-style composite TV output. Pseudo color and high-color mode are used for graphics data. The pseudo color modes include 4-clor, 16-color and 256-color mode, while the high color mode is 565. Furthermore, W90221 supports the opaque function to save data bandwidth. This Display Controller includes the following modules to complete its functions**: FIFO and Flicker-Free filter, Color Space Conversion, Overlay Control, Tune, CRTC, WIN-KEY, TV-Encoder**

Block Diagram :



Features :

- Graphic Accelerator
  - Build-in a hardware cursor with resolution up to 64x64x2

- Video Accelerator
  - Build-in buffers supporting YUV 4:2:2 video-in data
  - Video-in may be 8-bit, 4 type of YUV component sequence are selectable for video-in
  - Arbitrary-scaling-down and duplicated-scaling-up for video-in overlapee in graphic display
  - 2D bi-linear interpolation scaling-up for full-screen video display
  - On chip CCCIR 601 YUV to RGB color space converter (CSC)

- Video Overlay Logic
  - Provide color key and window key
  - Full screen display switch for graphics and video data

- Display Interface
  - Supports analog monitor up to 800x600 resolution, high color, non-interlace, 40MHz pixel clock and 60 frames/sec
  - On chip TV encoder supporting NTSC or PAL system
  - RCA-style composite video and S-Video
  - Triple 8-bit RGB video DACs are integrated
  - 3-line flicker free filter

Related Pins :

A. Video-in interface :

- VIN[0:7] (in/out) :
  During normal mode, this bus **inputs** 8-bit digital components of YCbCr 4:2:2 video-in data from external video controller (ex. TV decoder or MPEG decoder).
  During test mode, this bus **outputs** 8-bit digital components of YCbCr 4:2:2 video-out generated by internal video accelerator (VA).

- HSI (input) :
  Horizontal Sync of video-in frames. The content of VAconf[?] determines the polarity of this signal.

- VSI (input) :
  Vertical Sync of video-in frames. The content of VAconf[?] determines the polarity of this signal.

- VCLK (input) : (mentioned in section 5.1 "Clock Module and PLL")

B. Display interface :

- CP2/C/Blue (output) :
  During composite video mode (NTSC, PAL), this pin is analog "composite video" output.
  During S-Video mode, this pin is analog "chrominance" output.
  During monitor mode, this analog output supplies current corresponding to the "blue" intensity of the pixel being displayed.
  (To maintain IBM VGA compatibility, R-G-B outputs are typically terminated to monitor's ground with a 75 omv 2% resistor. This resistor, in parallel with the 75 omv resistor in the monitor, will yield a 37.5 omv impedance to ground. For a full-scale voltage of 700 mV, full-scale current output will be 18.7mA.)

- CP1/Y/Green (output) :
  During composite video mode (NTSC, PAL), this pin is analog "composite video" output.
  During S-Video mode, this pin is analog "luminance" output.
  During monitor mode, this analog output supplies current corresponding to the "green" intensity of the pixel being displayed.

- CP0/Red (output) :
  During composite video mode (NTSC, PAL), this pin is analog "composite video" output.
  During S-Video mode, this pin left no connection.
  During monitor mode, this analog output supplies current corresponding to the "red" intensity of the pixel being displayed.

- HSO (output) :
  Horizontal Sync of the displayed graphic output. The content of VAconf[?] determines the polarity of this signal.

- VSO (output) :
  Vertical Sync of the displayed graphic output. The content of VAconf[?] determines the polarity of this signal.

C. DAC interface :

- VREF (analog Out) : (optional)
  Voltage Reference Out; Bypass and decouple the voltage reference with 0.1uF ceramic capacitor to the TVDD.
  The decoupling capacitor shall be as close to the chip as possible. This pin as well as "COMP" are used to control the current of internal current sources are exactly equal to "Iref".

- COMP (analog Out) :
  Compensation pin. It shall be decoupled with a 0.1uF ceramic capacitor to TVDD. The decoupling capacitor shall be as close to the chip as possible.

- RSET (analog Out) :
  Current Source Adjusting Resistor. This pin is used to adjust the full scale current of TV's analog outputs. A resistor shall be connected between this pin and TVSS. (The DAC's "Iref" of current mirrors are adjusted by

this pin). The Iref is approximate to 1.16V/RSET.

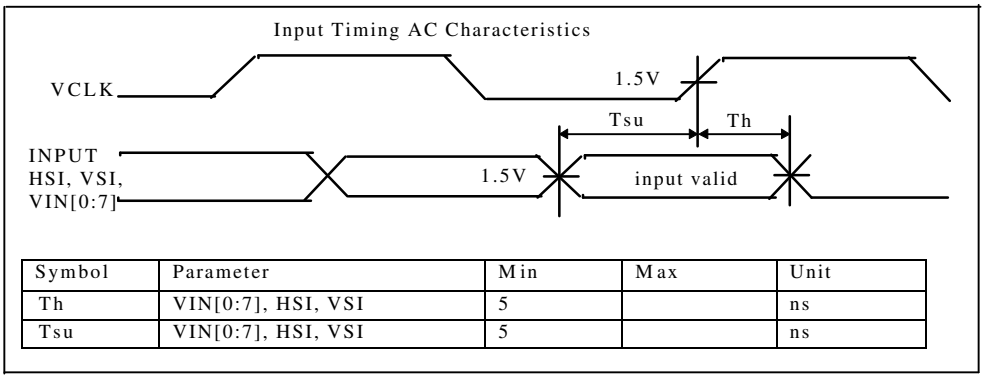
- EXTVREF (analog In) : (optional)
External Vref input. This signal supplies the DAC's "bandgap" output from a external 1.235V voltage source. A
0.1uF bypass capacitor should be always connected between this pin and TVDD. ("bandgap" is an voltage
stabilizer of voltage-reference-generator "Vref").

- TVDD, TVSS :
Dedicate power/ground pins for internal DACs.

Application Notes :

In the video pre-processing(VPRE) block that is designed to capture video image. A digital camera, or an NTSC/PAL camera connected to a TV decoder is fed into the W90221 in YCbCr 4:2:2 format through 8-bit data bus(VIN[0:7]). The input video is cropped and scaled, then display to output device. During operation mode, VCC[22] of VPRE must be set to zero to control data bus direction. And control data stream format by VCC[26:27] of VPRE.

The following figures show some typical timing diagrams of VPRE input pins :



Symbol	Parameter	Min	Max	Unit
Th	VIN[0:7], HSI, VSI	5		ns
Tsu	VIN[0:7], HSI, VSI	5		ns

In the video pos-processing(VPOST) block that is designed to support two kind of display devices, TV and Monitor. On TV Mode, when TVTWH[12:13] is set to 1X, W90221 operate on TV-system. Clock (DPCLK) is equal to 27MHz. On Monitor Mode, when TVTWH[12:13] is set to 0X, W90221 operate on Monitor mode. Clock (GFXCLK) is equal to 36.864MHz.

5.6 DMA CONTROLLER

Overview :
(Left for Blank)

Block Diagram :
(Left for Blank)

Features :

- flexible block-transfer mode and demand mode are supported
- provides 8-bit ecp-to-memory or memory-to-ecp transfer mode
- provides 8-, 16- and 32-bit memory-to-memory transfer modes
- DMA transfer between PCI memory to/from system memory are also support
- 4 words (16 bytes) memory burst-access; linear burst order
- build-in 4-words data FIFO to accelerate memory access
- the starting address of source and target shall be halfword boundary for 16-bit memory transfer and word boundary for 32-bit memory transfer

Related Pins : None

Operation Modes :
(Left for Blank)

5.7 PCI BRIDGE

Overview :
The W90221 host bridge provides a PCI bus interface that is compliant with the PCI local bus specification, revision 2.1, the implementation is optimized for high performance data streaming when the W90221 is acting as either the target or the initiator on the PCI bus.

Block Diagram :
(Left for Blank)

Features :
- Supports up to 4 external PCI bus master.
- Flexible programming external PCICLK delay reference to internal EXTCLK.
- Provides fix/rotate priority abitration.
- Provides configuration read/write, I/O read/write, memory read/write access.

Related Pins :

- PCLK (in) :
PCLK provides timing for all transactions on PCI and is an input to every PCI device.

- FRAME# (inout) :
FRAME# is an output when W90221 acts as an initiator on the PCI bus, FRAME# is asserted to indicate a bus transaction is beginning. While FRAME# is asserted, data transfer continue.
When FRAME# is deasserted, the transaction is in the final data phase or has completed.
FRAME# is an input when W90221 acts as a PCI target.

- PDA[31:0] (inout) :
These signals are connected to the PCI address/data bus. Address is driven by W90221 with FRAME# is asserted, data is driven or received in the following clocks. when W90221 acts as a target on the PCI bus, the AD[31:0] signals are inputs and contain the address during the first clock of FRAME# assertion and input data(writes) or output data(reads) on sebsequent clocks.

- COMBE#[3:0] (inout) :

PCI bus command and byte enable signals are multiplexed on the same pins. During the address phase of a transaction, COMBE#[3:0] define the bus command. During the data phase, COMBE#[3:0] are used as byte enables. The byte enables deterrmine which byte lanes carry meaningful data. The provided bus command encoding and types are listed below :

COMBE#[3:0]	Command type
0010	I/O read
0011	I/O write
0110	Memory read
0111	Memory write
1010	Configuration read
1011	Configuration write

- IRDY# (inout) :
IRDY# is an output when W90221 acts as an initiator on the PCI bus and an input when W90221 acts as a PCI target. The assertion of TRDY# indicates the current PCI bus initiator can complete the current data phase of the transaction.

- TRDY# (inout) :
TRDY# is an input when W90221 acts as an initiator on the PCI bus and an output when W90221 acts as a PCI target. The assertion of IRDY# indicates the current PCI target can complete the current data phase of the transaction.

- STOP# (inout) :
STOP# is an input when W90221 acts as an initiator on the PCI bus and an output when W90221 acts as a PCI target. STOP# is used for disconnect, retry, and abort sequences on the PCI bus.

- DEVSEL# (inout) :
Device select, when asserted, indicates that a PCI target device has decoded its address as the target of the current access. The W90221 asserts DEVSEL# based on the DRAM address range being accessed by a PCI initiator. As an input it indicates whether any device on the bus has been selected.

- PERR# (inout) :
PERR# indicates the current transaction has data parity error occurs. it is an input when W90221 acts as an PCI initiator and the current transaction is write access or W90221 acts as an PCI target and the current transaction is read access. it is an output when W90221 acts as PCI initiator and the current transaction is read access or W90221 acts as an PCI target and the current transaction is write access. When PERR# asserted and Master 0 Latency Register bit 17=1, it will genwrate NMI (non-maskable interrupt).

- PPAR# (inout) :
PPAR# is driven by the W90221 when it acts as a PCI initiator during address and data phases for a write cycle, and during the address phase for a read cycle. PPAR is driven by the W90221 when it acts as a PCI target during each data phase of a PCI memory read cycle. Even parity is generated across PDA[31:0] and COMBE#[3:0].

- PREQ0# (in) :
PREQ0# is the PCI bus request signal used as an input to indicate the arbiter that this master 0 desires use of the bus.

- PREQ1# (in) :
PREQ1# is the PCI bus request signal used as an input to indicate the arbiter that this master 1 desires use of the bus.

- SERR# (in) :
SERR# is the system error reporting, if SERR# asserted and Master 0 Latency Register bit16 =1, it will genertae a NMI (non-maskable interrupt).

- INTA# (in) :
Interrupt A is used to request an interrupt.

- INTB# (in) :
Interrupt B is used to request an interrupt.

- INTC# (in) :
Interrupt C is used to request an interrupt.

- PCIRST# (output) :
PCIRST# is used to reset PCI device.

- GNT0# (output) :
GNT0# is the PCI bus grant output signals generated by the internal PCI arbiter.

- GNT1# (output) :
GNT1# is the PCI bus grant output signals generated by the internal PCI arbiter.

- GPIO[16:15] (output) :
If ECP not enable, the GPIO[16:15] indicates PCI bus grant output GNT[3:2]#.

- INTD# (output) :
It has no meaning on PCI bus, when asserted, it indicates AIO global chip select.

Operation Modes :

- DX3 Mode : Once MD[26:27] is set to 00 during power on reset, the PCICLK will operate at CPUCLK/3 frequency.
- DX4 Mode : Once MD[26:27] is set to 01during power on reset, the PCICLK will operate at CPUCLK/4 frequency.
- DX5 Mode : Once MD[26:27] is set to 10 during power on reset, the PCICLK will operate at CPUCLK/5 frequency.
- DX6 Mode : Once MD[26:27] is set to 11 during power on reset, the PCICLK will operate at CPUCLK/6 frequency.

- PCICLK skew control : Bit 17 ~ 20 of "Master 2 Latency Register" are used to adjust the skew of PCICLK so as to to make all other PCI control signals get enough setup and hold time releated to PCICLK.

Some typical value are suggested as following :

> For DX3 : REG2[17:20] = 0100
> For DX4 : REG2[17:20] = 0111
> For DX5 : REG2[17:20] = 1001
> For DX6 : REG2[17:20] = 1011

5.8 AIO BUS CONTROLLER

Overview:
(Left for Blank)

Block Diagram:

(Left for Blank)

Features:
- One **16M** space for memory device and one **64K** space for IO device
- 8-bit or 16-bit IO access
- 8-bit memory write, 32-bit data-memory read and 4*32-bit code-ROM burst read
- Memory space (ROM/Flash) are always non-cacheable except code-ROM
- Provide no DMA transferring
- Programmable command wait states, set-up and hold time for all access

Related Pins:
AIO bus is an ISA-like bus and shares the existing 37 pins with PCI bus bridge. When AIO bus is enable, Only PCI
interrupt requests INTA#, INTB# and INTC# are available. The INTD# has been used as AIO's global chip select in that case.

- PDA[31:24]/**XA[8:15]/XD[15:8]** (inout) :
  PCI cycles: Serve as highest byte of PCI 32-bit address/data bus.
  AIO memory cycle: Serve as highest byte of 24-bit address lines (XA[8:31]) during AIO memory cycles.
  AIO IO cycles: Serve as high byte of 16-bit data lines (XD[15:0]) during AIO IO cycles.

- PDA[23:8]/**XA[16:31]** (inout) :
  PCI cycles : Serve as bits 16-31 of PCI 32-bit address/data bus.
  AIO cycles : Serve as lower 16-bit of 24-bit address lines (XA[8:31]) during all AIO cycles.

- PDA[7:0]/**XD[15:8]** (inout) :
  PCI cycles : Serve as lowest byte of PCI 32-bit address/data bus.
  AIO memory cyc. : Serve as the 8-bit data lines during AIO memory cycles.
  AIO IO cycles. : Serve as low byte of 16-bit data lines (XD[15:0]) during AIO IO cycles.

- COMBE[3]/**AIOCS#** (inout) :
  PCI cycles : Bit-3 of command/byte bus
  AIO cycles : AIO chip-select for its IO devices

=

- COMBE[2]/**XROMCS#** (inout) :
  PCI cycles : Bit-2 of command/byte bus
  AIO cycles : AIO chip-select for its memory devices

- COMBE[1]/**XWR#** (inout) :
  PCI cycles : Bit-1 of command/byte bus
  AIO cycles : Asserted low, if INTD# is also low, indicating a AIO write command cycle is ongoing.

- COMBE[0]/**XRD#** (inout) :
  PCI cycles : Bit-0 of command/byte bus
  AIO cycles : Asserted low, if INTD# is also low, indicating a AIO read command cycle is ongoing.

- INTD#/**XGLBCS#** (output) :
  PCI cycles : If AIO is enable, this signal shall not connect to any PCI bus master.
  AIO cycles : Asserted low indicating a AIO command cycle is ongoing.

Application Notes :
AIO bus is designed to connect ISA-like, low speed devices such as code-ROM, Flashs and 8-/16-bit IO devices. The AIO controller itself is a PCI slave device, if the address and access type of any PCI cycle match the AIOBASE or XMBASE[8:15] of AIO, the AIO controller responds the DEVSEL# and TRDY# to PCI bridge and generate correspond AIO bus signals to AIO devices in the "data phase" of current PCI cycle.

XMBASE[7] will be set right from chip reset, all PCI cycles will be treated as AIO access and all code read (PCI cycles) will return data from AIO bus. After the memory (XMBASE) and IO (AIOBASE) have been properly configured, XMBASE[7] shall be set logic low immediately to avoid possible wrong response from AIO controller.

Because XROMCS#, AIOCS#, XRD# and XWR# are shared the same pins with COMBE[0:3] of PCI bus, they might toggle during any PCI cycles. It is necessary to OR these control signals with INTD #, which dedicately serve as "AIO global chip-select", before they reaching the AIO devices.

The following figures show some typical timing diagrams of AIO command cycles :

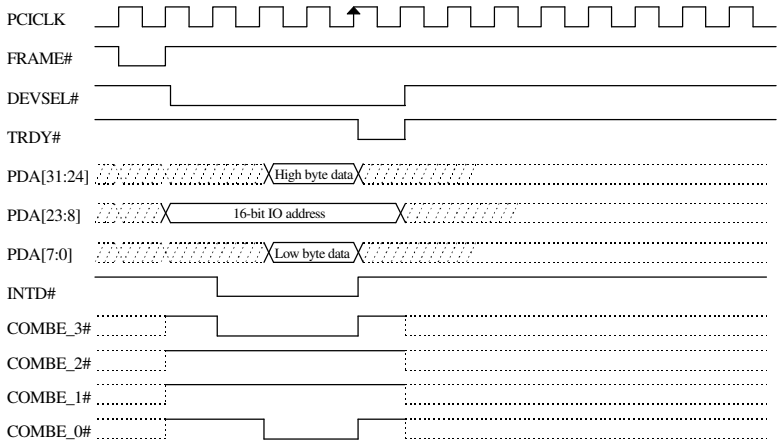


Fig 5.8.1 Fastest XIO IO read cycle (0 wait)

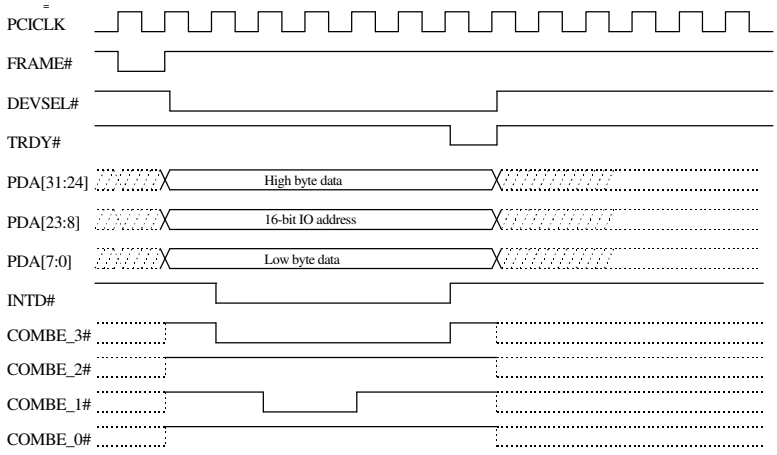


Fig 5.8.2 Fastest XIO IO write cycle (0 wait)

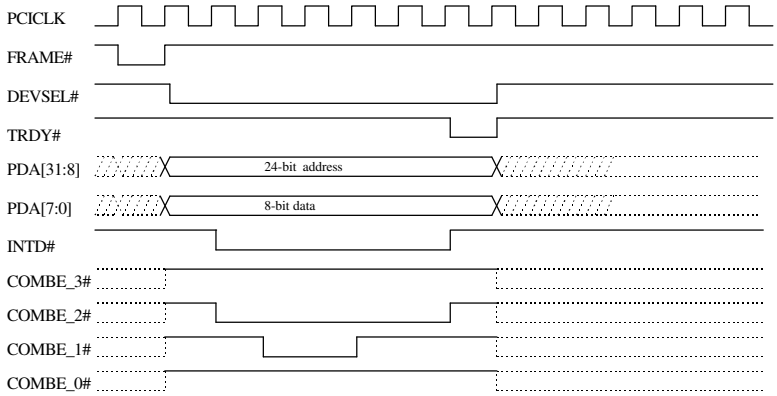


Fig 5.8.3 Fastest 8-bit memory write (0 wait)

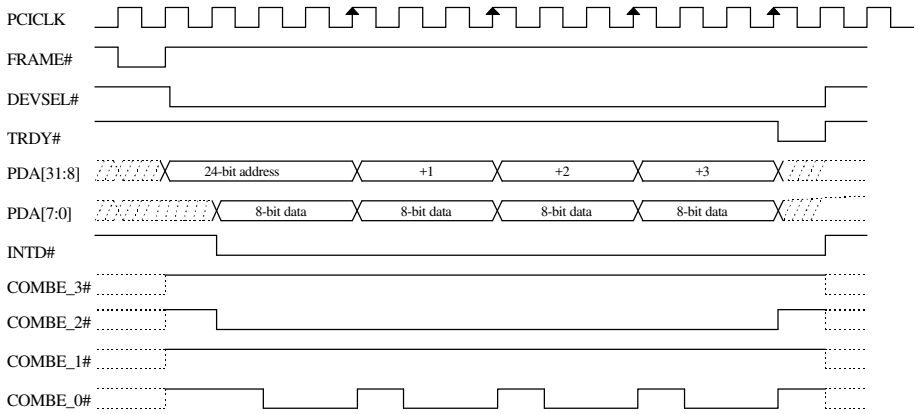


Fig 5.8.4 Fastest 32-bit memory read (0 wait)

As for 4-word code ROM **burst read**, the timing diagram is similar to Fig5.8.4 except there will be consective **16**
COMBE_0# (XRD#) command pulses within one chip select (INTD#, COMBE_2#).

5.9 PARALLEL PORT INTERFACE

Overview :
(Left for Blank)

Block Diagram :

(Left for Blank)

Features :
- supports all IEEE P1284 transfer modes including :
- Compatible (centronic) mode (forward channel)
- Nibble mode (reverse channel, compatible with all existing PC hosts - relies on **software control**)
- Byte mode (reverse channel, compatible with IBM PS/2 host)
- EPP mode (bi-directional half-duplex channel - relies on **software control**)
- ECP mode (fast bi-directional half-duplex channel)
- Host-side design
- Provide a special operation mode to emulate peripheral-side centronic device
- Build-in one 16bytes FIFO to accelerate ECP mode and centronic forward transfer
- Provide DMA capability to accelerate moving data from parallel port interface to system memory
- ECP mode is also including :
- High performance half-duplex forward and reverse channel
- Interlocked handshake, for fast reliable transfer
- Forward "channel-addressing/command transfer" for low-cost peripherals
- Support reverse RLE decompression
- Peer-to-peer capability

Related Pins :
All 17 PPI interface signals share pins with GPIOs. When T_ENECP (port 0xf000003e , bit 4) is set, the corresponding
GPIOs are redefined as following :

- GPIO[16]/**nStrobe** (output) :
  Compatible Mode : Set active low to transfer data into peripheral device's input latch
  ECP Mode : Used in a closed-loop handshake with "Busy" to transfer data or address information from host to peripheral device.

- GPIO[15]/**nAutoFd** (output) :
  Compatible Mode : Set low by host to put some printers into auto-line feed mode. May also be used as a ninth data, parity, or command/data control bit.
  ECP Mode : The host drives this signal for flow control in the reverse direction. It is used in an interlocked handshake with "nAck". "nAutoFd" also provides a ninth data bit used to

determine whether command or data information is present on the data signals in the
forward transferring.

- GPIO[8]/**nInit** (output) :
Compatible Mode : Pulsed low in conjunction with "nSelectIn" active low to reset the interface and force a
return to compatible mode idle state.
ECP Mode : This signal is driven low to place the channel in the reverse direction. While in this mode,
the peripheral is only allowed to drive the bi-directional data signals when "nInit" is low
and "nSelectIn" is high.

- GPIO[9]/**nSelectIn** (output) :
Compatible Mode : Set low by host to select peripheral device.
ECP Mode : Driven high by host while in ECP mode. Set low by host to terminate ECP mode and
return the link to the compatible mode.

- GPIO[17]/**nAck** (input) :
Compatible Mode : Pulse low by the peripheral device to acknowledge transfer of a data byte from the host.
ECP Mode : Used in a close-loop handshake with "nAufoFd" to transfer data during reverse transferring.

- GPIO[18]/**Busy** (input) :
Compatible Mode : Driven high to indicate that the peripheral device is not ready to receive data.
ECP Mode : The peripheral device uses this signal for flow control in the forward transferring. "Busy"
also provides a ninth data bit used to determine whether command or data information is
present on the data signals in the reverse direction.

- GPIO[11]/**PError** (input) :
Compatible Mode : Driven high to indicate that the peripheral device has encountered an error in its paper path
(ex. paper empty). Peripherals shall set "nFault" low whenever they set "PError" high.
ECP Mode : Peripherals drive this signal low to acknowledge "nInit". The host relies upon "PError" to
deterine when it is permitted to drive the data signals.

- GPIO[10]/**Select** (input) :
Compatible Mode : Set high to indicate that the peripheral device is on-line.
ECP Mode : Used by peripheral to reply to the requested extensibility byte sent by the host during the
negotiation phase.

- GPIO[12]/**nFault** (input) :
Compatible Mode : Set low by peripheral device to indicate that an error has occured.
ECP Mode : Set high to acknowledge 1284 compatibility during negotiation phase. During ECP mode

=

the peripheral may drive this pin low to request communications with the host. This signal would be typically used to generate an interrupt to the host. This signal is valid in both forward and reverse trnasfers.

- GPIO[0:7]/**ED[0:7]** (in/out) :
8-bit bus used to hold data, address or command information in all modes. The bit 0 is the most significant bit.

Operation Modes :
(Left for Blank)

5.10 UART

Overview :
The W90221 contains two Universal Asynchronous Receiver/Transmitter (UART) ports, one of them provides complete MODEM-control and serial transfermation capabilities, whereas the other one provides only serial transfermation capability. The UART performs serial-to-parallel conversion on data characters received from a peripheral device such as MODEM, and parallel-to-serial conversion on data characters received from the CPU. One 16 bytes transmitter FIFO (TX-FIFO) and one 16 bytes (plus 3 bits of error data per byte) receiver FIFO (RX-FIFO) have been built in to reduce the number of interrupts presented to the CPU. The CPU can read the complete status of the UART at any time during the functional operation. Status reported includes error conditions (parity, overrun, framing, or break interrupt) and states of TX-FIFO and RX-FIFO.

Block Diagram :

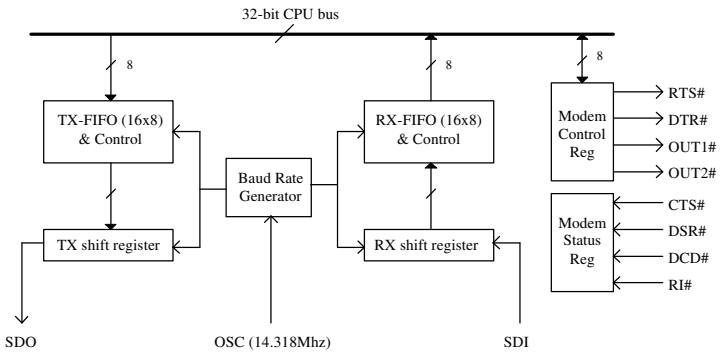


Fig 5.1.5-1 UART Block Diagram

Features :
- transmitter and receiver are each buffered with 16 bytes FIFO's to reduce the number of interrupts presented to the CPU
- MODEM control functions (CTS, RTS, DSR, DTR, RI and DCD)
- Fully programmable serial-interface characteristics :

-- 5-, 6-, 7-, or 8-bit characters
-- even, odd, or no-parity bit generation and detection
-- 1-, 1&1/2, or 2-stop bit generation
-- baud rate generation

- line break generation and detection
- false start bit detection
- full prioritize interrupt system controls
- loop back mode for internal diagnostic testing

Related Pins :
(COM0)
- SIN0 (input) : Serial data input from peripheral device or MODEM
- SOUT0 (output) : Serial data output to peripheral device or MODEM
- CTS1# (input) : Clear to send signal
- DSR1# (output) : Data set ready
- DTR1# (input) : Data terminal ready
- RTS1# (output) : Request to send
- DCD1# (input) : Data carrier detect
- RI1# (output) : Ring indciator

(COM1)
- SIN1 (input) : Serial data input from peripheral device or MODEM
- SOUT1 (output) : Serial data output to peripheral device or MODEM

Operation Modes :
- Interrupt Mode operation :
  A. Receiver control :
    - Set FCR[0:1] to select a proper receiver threshold level and then turn on "receiver data available interrupt" (Irpt_RDA) by set IER[7] to logic 1.
    - The Irpt_RDA will be triggered when the receiver FIFO (RX-FIFO) has reached its programmed trigger level, and it will be cleared as the available data in RX-FIFO drops below the trigger level.
    - As Irpt_RDA occured, the corresponding IIR bits will be set to inform the software application that data in RX-FIFO has reached programmed threshold level.
    - If the received data has any errors, the "line status interrupt" (Irpt_RLS) will occur and has higher priority than Irpt_RDA.
    - If "time out interrupt" (Irpt_TOR) is enable by set IER[7] and TOR[0] to logic 1s. The Irpt_TOR will occur, if the following conditions exist :
      - at least one character is in RX-FIFO.
      - RX-FIFO is not received any data or accessed by CPU from the most recent serial character received, and the time period, counting by baud rate bit clock, has exceeded the value being programmed in TOR[1:7].
    - The Irpt_TOR and the time-out counter will be cleared as the CPU reads one character from RX-FIFO.
    - The time-out counter is reset after a new character is received or after the CPU reads the RX-FIFO.

  B. Transmitter control :

- Set IER[6] to logic 1 to enable "transmitter empty interrupt" (Irpt_THRE) before transmitter operation.
- Once the transmitter FIFO (TX-FIFO) is empty, the Irpt_THRE is triggered and the corresponding IIR bits are set to inform the CPU to fill the TX-FIFO (maximum 16 bytes of characters).
- The Irpt_THRE is reset after the CPU reads the IIR (IIR[4:7] must be 4'b0010 at that time) or writes a character into TX-FIFO.
- Irpt_RDA and Irpt_TOUT has the same interrupt priority (2nd priority) while Irpt_THRE has a lower priority (3rd priority).

- Polled Mode operation : (refer to "LSR" register discriptions located on Section 5.2.5)
  - No interrupts need be enabled at this mode, the CPU always polls the LSR to check COM port status before taking any actions.
  - LSR[7] will be set as long as there is at least one byte in the RX-FIFO, and it is cleared if the RX-FIFO is empty.
  - LSR[3:6] will specify error(s) status which is handled the same way as in the interrupt mode operation, the IIR[4:7] is not affected since no interrupt is enabled.
  - LSR[2] will indicate when the TX-FIFO is empty.
  - LSR[1] will indicate that both TX-FIFO and shift register are empty.
  - LSR[0] will indicate whether there are any errors in the RX-FIFO.

5.11 SYNCHRONOUS SERIAL INTERFACE (SSI)

Overview :

The SSI module within W90221 contains holding registers, shift registers, and other logic to support a variety of serial data communications protocols and provide a direct connection to external audio/telephony codec devices.

Two 48 halfwords fifos, the transmitter fifo and receiver fifo, have been implented to accelerate both transmittion and receiving operations. These two fifos can be configured as 48 halfwords or 24 words depth depending on the data word length.

Block Diagram :

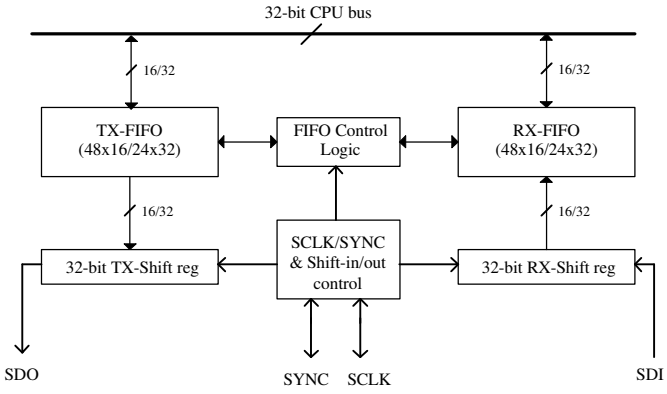


Fig 5.1.6-1 SSI Block Diagram

Features :

- supports "long framing" and "short framing" (synchronous, frame-based protocol)
- provides "master mode" and "slave mode"
- build-in two 48x16 (or 24x32) data fifo to accelerate transmit/receive operation
- Programmable data bits per one frame (sampling rate) : 1 ~ 256 bits/frame
- Programmable data bits per word (resolution of each sampling) : 1 ~ 32 bits/word
- Programmable multi-word (per frame) transfer : 1 ~ 16 words/frame

Related Pins :

- SDI (input) : This pin contains the input data shifted from external audio/telephony codec devices
- SDO (output) : This pin contains the output data shifted to external audio/telephony codec devices
- SYNC (in/out) : This pin is the frame synchronization signal between SSI and codec devices. The SYNC may be input or output depending on SSI operated in slave- or master-mode respectively.
- SCLK (in/out) : This pin is the serial bit clock between SSI and codec devices. Likewise, The SCLK may be input or output depending on SSI operated in slave- or master-mode respectively.

Operation Modes :

- Master Mode : Once CFGH[2] is set to logic 1 and MD[25] is pull high, SSI is operated in master mode, and the SYNC (determines the sampling rate) and SCLK is drived by SSI module to external codec devices.

SCLK frequence = EXTCLK/[2*(CFGL[8:15] + 1)] (5.1.6a)
SYNC period = SCLK * (CFGL[0:7] + 1) (5.1.6b)

- Slave Mode : Once CFGH[2] is set to logic 0 and MD[25] is pull down, SSI is operated in slave mode, the SCLK and SYNC are drived externally (may be from codec devices). So the sampling rate and SCLK frequence are determined by external devices, however software driver still need to properly set "serial data bit length" (CFGH[8:10]) as well as "data words per frame" (CFGH[12:15]) to make SSI module working correctly.

- Loop mode : This mode (CFGH[1] =1) aims at selftesting. When this bit is set, serial data-out "SDO" is connected to serial data-in "SDI" internally and SDO pin fixed at logic 0 state. Besides, if **Loop** and **Master** mode are chose concurrently, SSI module will not issue SYNC until TX-FIFO contains at least one data word.

- Long Framing : When CFGH[3] is set to logic 1, SSI is operated in long framing mode. The following features are included in long framing mode :
consists of the following features.
- The SSI module always samples receive date (SDI) on the falling edge of SCLK, whereas always pushes transmit data (SDO) on the rising edge of SCLK.

- The frame sync (SYNC) is asserted immediately as the first bit of transmit and receive data.
- The frame sync (SYNC) is asserted for one "serial word length" which determined by CFGH[8:11].

Serial word length = CFGH[8:11] + 1 (5.1.6c)

- The frame sync rate (sampling rate) and SCLK frequence follow eq (5.1.6b) and (5.1.6a) respectively on master mode and determined by external devices on slave mode.
- The transmit FIFO and receive FIFO is configued as 48x16 if "serial word length" <= 16, and will be configured as 24x32 if "serial word length" > 16.
- The shifting data bits on SDI and SDO are always MSB first.
- If serial word length is not 16 or 32, it is software responsibility to left(MSB) justify the transmit data words before writing it to transmit FIFO, the received data before being written into receive FIFO is righ(LSB) justified automatically by SSI module where the unfilled MSBs are catneted with logic 0s.
- SSI module always shifts out logic 0s on each frame sync if transmit FIFO is empty at that time.
- A receiver FIFO interrupt will be asserted (when RX-FIFO interrupt is enable) if the received data words exceeds the receive FIFO's threshold level. Likewise, a transmitter FIFO interrupt will be asserted (when TX-FIFO interrupt is enable) if the available data words in transmit FIFO is lower than its threshold level.
- Fig 6.1.5-2 shows a standard long framing transfer where serial word length is 3 (CFGH[8:11] = 2), words per frame is 3 (CFGH[12:15]=2) and bits per frame is 9 (CFGL[0:7] = 10).

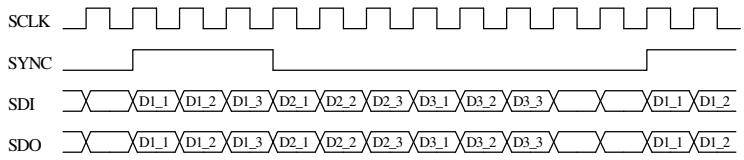


Fig 5.1.6-2 SSI long framing transfer

- Short Framing : When CFGH[3] is set to logic 0, SSI is operated in long framing mode. The following features are included in short framing mode consists of the following features.
- The frame sync (SYNC) is asserted for one SCLK immediately before the first bit of transmit and receive data.
- The frame sync (SYNC) is asserted for one SCLK period.
- All other features are the same as long framing mode.
- Fig 6.1.5-3 shows a standard short framing transfer where serial word length is 3 (CFGH[8:11] = 2), words per frame is 3 (CFGH[12:15]=2) and bits per frame is 9 (CFGL[0:7] = 10).

=

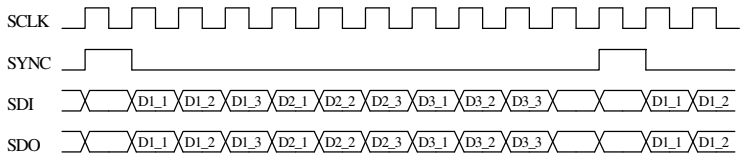
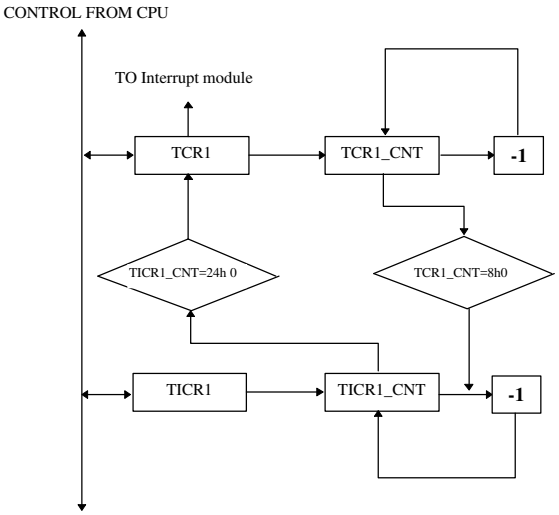


Fig 5.1.6-3 SSI short framing transfer

5.12 TIMER CHANNELS

Overview :
Two 24-bit decrementing timers are implemented, corresponding to the TCR1, TICR1 and TCR2, TICR2 independently. When the timers' interrupt enable bit is set high and the counter decrements to zero, the timer will assert its interrupt requext signal. When a timer reaches zero, the timer hadrware reloads the counter with the value from the timer initial counter register and continues decrementing.

Block Diagram :



***Note :** The above block diagram is timer channel 1. Block diagrams of these two timer channels are the same.

Features :
- Two 24-bit decremental timer channels with individual interrupt requests
- Programmable timer clocks for each channels, the clock range is OSC ~ OSC/8'hFF
- maximum uninterrupted time or timeout = 5 minutes (if OSC = 14.318MHz)
- Typical OSC frequence is 14.318MHz.

Related Pins : (None)

Operation Modes :

The following figures show some typical timing diagrams of TIMER write cycles

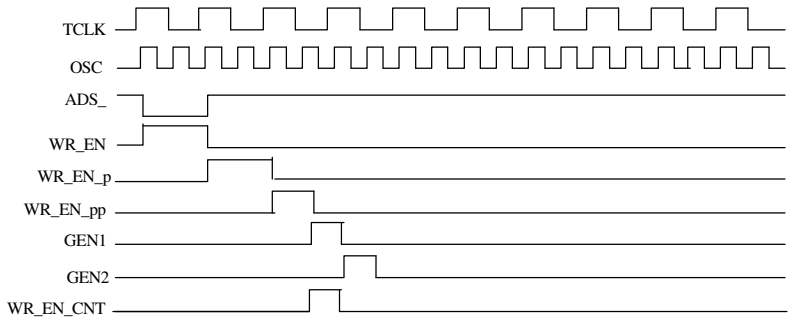


Fig5.12.1 Timer register write commandwhen the osilattor frequence is faster the cpu frequency

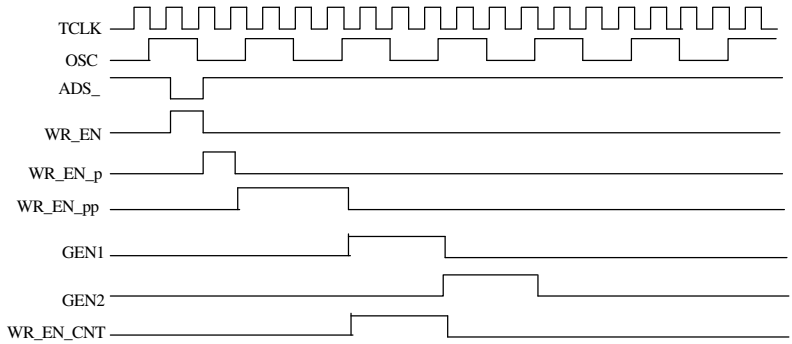


Fig5.12.1 Timer register write command when the osilattor frequence is slower the cpu frequency

***Note :**
WR_EN : The write enable signal for timer register reference the TCLK clock
WR_EN_CNT : The write enable signal for timer count register reference the OSC clock.

6 REGISTER DEFINITIONS

6.0 GPIO REGISTERS

There are six registers included in the Pio module. The IO address map is allocated from 0xf0000050 to 0xf0000064.

Table 6.0.1 GPIO Register Map
(IO base (BA) : 0xf0000000)

Port Addr.	Symbol	Access	Description
BA + 0x50	PIEN	R/W	Interrupt Enable Register
BA + 0x54	POEN	R/W	Output Enable Register
BA + 0x58	PODATA	R/W	Output Data Register
BA + 0x5C	PINT	R/W	Interrupt Request Register
BA + 0x60	PIDATA	R	Input Data register
BA + 0x6C	CLKREG	R/W	Debounce Clock Select Register

GPIO Interrupt Enable Register (PIEN)

Port address : 0x00000050 Read/Write Power-on Default : 32h0000_0000

0 15	16 30	31
Reserved	PIEN	Reserved

Bit 0_15 : Reserved

Bit 16_30 : PIO Interrupt Mask bits :
These 15 bits serve as interrupt enable bits of GPIO[0:14] respectively, when GPIOs are programmed
as input mode. Setting these bits to logical 0s, the releated GPIO interrupt requests are pended in "PIO
interrupt request" register (PINT).

Bit 31 : Reserved

GPIO Output Enable Register (POEN)

Port address : 0x00000054 Read/Write Power-on Default : 32h0000_0000

0 15	16 30	31
Reserved	POEN	Reserved

Bit 0_15 : Reserved

Bit 16_30 : GPIO pin output-enable :
Setting any of these bits to logical high, the corresponding GPIO[0:14] pins will act as output pin.

Bit 31 : Reserved

GPIO Data-Out Register (PODATA)

Port address : 0x00000058 Read/Write Power-on Default : 32h0000_0000

0 15	16 30	31
Reserved	PODATA	Reserved

Bit 0_15 : Reserved

Bit 16_30 : GPIO data-out bits :
The logical state of these bits will be echoed on corresponding GPIO[0:14] pins, if any of GPIO pins
are programmed as output mode.

Bit 31 : Reserved

GPIO Interrupt Request Register (PIEN)

Port address : 0x0000005C Read/Write Power-on Default : 32h0000_0000

0 15	16 30	31
Reserved	PINT	Reserved

Bit 0_15 : Reserved

Bit 16_30 : GPIO Interrupt Request :
When GPIO is programmed as input pin, any transition in GPIO pins (**a recognized logic state shall keep**

stable for at least 2 debounce clock (TCLK_BUN)) will set releated bits in this register to logical high.

Besides, GPIO module will not issue interrupt request to CPU host until the same bits of PIEN and PINT are

both set high.

Bit 31 : Reserved

GPIO Input Data Register (PIDATA)

Port address : 0x00000060 Read Power-on Default : 32h0000_0000

0 15	16 30	31
Reserved	PIDATA	Reserved

Bit 0_15 : Reserved

Bit 16_30 : GPIO Pin Status :

These 15 bits always echo GPIO[0:14] pin status no matter the GPIOs are programmed as input or output

pins. All input ports of GPIO[0:14] are debunced first by TCLK_BUN before they be echoed by these bits.

Bit 31 : Reserved

Debounce Clock Select Register

Port address : 0x00000064 Read/Write Power-on Default : 32h0000_0000

0 28	29 31
Reserved	CLKREG

Bit 0_28 : Reserved

Bit 29_31 : Debounce Clock Rate Selector

These 3 bits are used to select the debounce circuit clock rate. The relationship of system clock (TCLK) and

debounce clock (TCLK_BUN) according to these 3 bits are as following :

TCLK_BUN = TCLK/CLKREG[29:31]

Each of 15 general purpose GPIO ports can be programmed as input or output port independently. Each port can generate positive or negative edge interrupt, and contains of a bi-directional buffer connected to the appropriate W90221 pin, the output signal from the input buffer is routed directly to a debounce circuit. This circuit performs 3 TCLK_BUN (program by CLKREG register) debounce of the input signal. Reading a specific bit location within the IO Data Input Register returns the logic state of the respective general purpose IO pin, regardless of whether that pin is configured as an output or input. If the pin is configured as an output an input, the value read is the logic state of the pin as driven by W90221 pin .

6.1 MEMORY CONTROLLER REGISTERS

There are 5 16-bit registers and 1 22-bit IO base register included in the memory (DRAM) controller. Access to these DRAM's registers are through a "IO base content + offset" port. Access to "IO base" register is through 0xf0000000 port. The memory controller supports **EDO** and **Synchronous** DRAM.

Table 6.1 : MEMC Register Map
(IO base (BA) : 0xf0000000)

Offset	Symbol	Access	Description
0x30h		R/W	DRAM bank 0 configuration register [0:15]
0x32h		R/W	DRAM bank 0 base register [0:15]
0x34h		R/W	DRAM bank 1 configuration register [0:15]
0x36h		R/W	DRAM bank 1 base register [0:15]
0x38h		R/W	DRAM control register[0:15]
0x3ah		R/W	DRAM timing register 0
0x3ch		R/W	DRAM timing register 1
0x3eh		R/W	DRAM timing register 2
0xf000000h		R/W	IO base address[0:21]

DRAM Bank Configuration Register ( )

Index : 0x30h,0x34h Read/Write Power-on Default : --

0	1	2	3	4	5	6	7
DRAM size			DRAM page size			COMPbk	BKen

8	9	10	11	12	13	14	15
Reserved							

Bits 0-2 Size of DRAM bank 0

Bits[0:2]			Size of DRAM type
0	0	0	1M
0	0	1	2M
0	1	0	4M
0	1	1	8M
1	0	0	16M
1	0	1	Reserved
1	1	0	Reserved
1	1	1	Reserved

The following table defines how CPU address bus map to DRAM address :

BA0 BA1

Total	Type	R x C	R/C	MA0	MA1	MA2	MA3	MA4	MA5	MA6	MA7	MA8	MA9	MA10	MA11	MA12	MA13	MA14
4M*	1Mx4	)x10		I7	I6	I5	14	I3	I2	I1	I0	I9	I8		'		·1	'
			C	A29	A28	A27	A26	A25	A24	A23	A22	A21	A20	A19	A18		A21	A7
;M*	/Ix4(x16)	x11		I7	I6	I5	14	I3	I2	I1	I0	A8	I8	A9	'		·1	'
			C	A29	A28	A27	A26	A25	A24	A23	A22	A21	A20	A19	A18		A21	A7
?M*	8Mx8	?x11		I7	I6	I5	14	I3	I2	I1	I0	A8	I8	A9	A7		·1	'
			C	A29	A28	A27	A26	A25	A24	A23	A22	A21	A20	A19	A18		A21	A7
IM*	16Mx4	?x12		I7	I6	I5	14	I3	I2	I1	I0	A8	A6	A9	A7		·1	'
			C	A29	A28	A27	A26	A25	A24	A23	A22	A21	A20	A19	A18		A21	A7
4M	1Mx16	Ix8		I7	I6	I5	14	I3	I2	I1	I0	I9	I8	?0	'		?1	'
			C	A29	A28	A27	A26	A25	A24	A23	A22	A21	A20	A19	A18		A21	A7
8M	2Mx8	Ix9		I7	I6	I5	14	I3	I2	I1	I0	I9	I8	A9	'		?0	'
			C	A29	A28	A27	A26	A25	A24	A23	A22	A21	A20	A19	A18		A20	A7
8M	2Mx32	Ix9		I7	I6	I5	14	I3	I2	I1	I0	I9	I8	A9	'		?0	'
			C	A29	A28	A27	A26	A25	A24	A23	A22	A21	A20	A19	A18		A20	A7
16M	4Mx4	x10		I7	I6	I5	14	I3	I2	I1	I0	A8	I8	A9	'		I9	'
			C	A29	A28	A27	A26	A25	A24	A23	A22	A21	A20	A19	A18		A19	A7
8M	/Ix32	Ix8		I7	I6	I5	14	I3	I2	I1	I0	I9	I8	A9	'		?1	?0
			C	A29	A28	A27	A26	A25	A24	A23	A22	A21	A20	A19	A18		A21	A20
16M	/Ix16	?x8		I7	I6	I5	14	I3	I2	I1	I0	A8	I8	A9	I9		?1	?0
			C	A29	A28	A27	A26	A25	A24	A23	A22	A21	A20	A19	A18		A21	A20
32M	8Mx8											A8		A9	A7			

		2x9		17	16	15	14	13	12	11	10		18				20	19
			C	A29	A28	A27	A26	A25	A24	A23	A22	A21	A20	A19	A18		A20	A19
32M	4Mx16	2x9		17	16	15	14	13	12	11	10	A8	18	A9	A7		20	19
			C	A29	A28	A27	A26	A25	A24	A23	A22	A21	A20	A19	A18		A20	A19
64M	8Mx4	2x10		17	16	15	14	13	12	11	10	A8	A6	A9	A7		19	18
			C	A29	A28	A27	A26	A25	A24	A23	A22	A21	A20	A19	A18		A19	A18
64M	8Mx8	2x10		17	16	15	14	13	12	11	10	A8	A6	A9	A7		19	18
			C	A29	A28	A27	A26	A25	A24	A23	A22	A21	A20	A19	A18		A19	A18

Note : * indicates EDO organization.

Bits 3-5 DRAM page size

Bits[3:5]			DRAM Page size
0	0	0	256 Bytes (x8)
0	0	1	512 Bytes (x9)
0	1	0	1K Bytes (x10)
0	1	1	2K Bytes (x11)
1	0	0	4K Bytes (x12)
1	0	1	Reserved
1	1	0	Reserved
1	1	1	Reserved

Bit 6 SDRAM component bank

0 = 2 banks

1 = 4 banks

Bit 7 DRAM bank enable

0 = disable

1 = enable

Bits 8-15 Reserved

These 2 16-bit registers defines the **configuration** of each DRAM banks'. offset 0x30h configure bank0, offset 0x34h configure bank1.

DRAM Bank Base Address Register ( )

Index : 0x32h,0x36h Read/Write Power-on Default : --

0	1	2	3	4	5	6	7
Base Address of bank BASEaddr[0:7]							

8	9	10	11	12	13	14	15
BASE addr[8:9]		Reserved					

These 2 16-bit registers defines the **most significant 10 bits** of each DRAM banks' base (bottom) address. The "DRAM base address" togather with the "DRAM size" (defined in DRAMconf0,1) construct the whole address range of each DRAM banks.

The base address of all two DRAM banks have no default value after power-on reset. It is software's responsibility to well program these registers before access system DRAM.

DRAM Control Register (DRAMctrl)

Index : 0x38h Read/Write Power-on Default : 0x0

0	1	2	3	4	5	6	7
DIMMbk	DRAMtp	ONsdram	TESTREF	PREchrg	MRS	REF	PDen

8	9	10	11	12	13	14	15
DISAF	MCLK ctrl select			MCLK data select			

Bit 0 banks per DIMM
0 = 1 bank
1 = 2 banks

Bit 1 DRAM type
0 = EDO
1 = SDRAM

Bit 2 On Board SDRAM
0 = DIMM
1 = On board

Bit 3 SDRAM RESET end select
This bit reserved for simulation only.

Bit 4 SDRAM precharge set
This bit set will issue a precharge command to SDRAM.

Bit 5 SDRAM mode register set
This bit set will issue a mode register set command to SDRAM.

Bit 6 SDRAM refresh cycle enable
This bit set will issue a refresh command to SDRAM.

Bit 7 SDRAM clock enable
0 = disable
1 = enable

Bit 8 Swap out 0xA0000 ~ 0xFFFFF
When this bit is set to a logic 1, address space 0xA0000 ~ 0xFFFFF will not be recognized as "system" DRAM space.

Bit 9-11 DRAM controller control signal clock skew adjustment reference CLKTREE MCLKo.

Bit 9-11			MCLK ctrl delay (xo02d2 delay)
0	0	0	0
0	0	1	2
0	1	0	4
0	1	1	6
1	0	0	8
1	0	1	10
1	1	0	12
1	1	1	14

Bit 12-14 DRAM controller data latch clock skew adjustment reference external MCLK.

Bit 12-14			MCLK data delay (xo02d2 delay)
0	0	0	0
0	0	1	2
0	1	0	4

0	1	1	6
1	0	0	8
1	0	1	10
1	1	0	12
1	1	1	14

RAM Timing Control Register_0 (DRAMTctrl0)

Index : 0x3ah Read/Write Power-on Default : 0x0

0	1	2	3	4	5	6	7
Refresh rate[0:2]			CAS Latency[0:1]		ACMDdly[0:1]		RASact[0]

8	9	10	11	12	13	14	15
RASact[1:2]		RAS precharge time[0:2]			DRAM cycle time [0:2]		

Bit 0-2 Refresh rate

Bit 0-2			Refresh rate
0	0	0	15us (default)
0	0	1	30us
0	1	0	60us
0	1	1	120us
1	0	0	240us
1	0	1	480us
1	1	0	960us
1	1	1	disable

Bit 3-5 CAS latency

Bit 3-4		CAS latency (MCLK)
0	0	Reserved
0	1	1
1	0	2
1	1	3 (default)

Bit 6 Active CMD delay

Bit 5-6	Active CMD delay (MCLK)
0 0	0
0 1	1
1 0	2
1 1	3 (default)

Bits 7-9 RAS# active pulse width

Bit 7-9	RAS# active pulse width (MCLK)
0 0 0	1
0 0 1	2
0 1 0	3
0 1 1	4
1 0 0	5
1 0 1	6
1 1 0	7
1 1 1	8 (default)

Bit 10-12 RAS# precharge time

Bit 10-12	RAS# precharge time (MCLK)
0 0 0	1
0 0 1	2
0 1 0	3
0 1 1	4
1 0 0	5
1 0 1	6
1 1 0	7
1 1 1	8

	(default)

Bit 13-15 DRAM cycle time

Bit 13-15			DRAM cycle time (MCLK)
0	0	0	2
0	0	1	3
0	1	0	4
0	1	1	5
1	0	0	6
1	0	1	7
1	1	0	8
1	1	1	9 (default)

DRAM Timing Control Register_1 (DRAMctrl1)

Index : 0x3c Read/Write Power-on Default : 0x0

0	1	2	3	4	5	6	7
CASPtime	CASRpul[0:2]			CASWpul[0:1]		CA2RA[0:1]	

8	9	10	11	12	13	14	15
RA2CD[0:1]		RAS to CAS delay[0:2]			LDI2ACT[0:2]		

Bit 0 CAS precharge time

0 = 1 MCLK

1 = 2 MCLK

Bits 1-3 CAS pulse width during Read cycle

(CAS# active-time during "**read cycle**")

Bit 1-3			CAS# active-time (MCLK)
0	0	0	1

0	0	1	2
0	1	0	3
0	1	1	4
1	0	0	5
1	0	1	6
1	1	0	7
1	1	1	8

Bits 4-5 CAS pulse width during write cycle

(CAS# precharge-time during "**write cycle**")

Bits 4-5		CAS# active-time (MCLK)
0	0	1
0	1	2
1	0	3
1	1	4

Bits 6-7 CAS# assertion to RAS# assertion (CA2RA[0:1])

These two bits determine the duration between CAS# assertion to RAS# assertion for CAS-before-RAS refresh cycle.

Bits 6-7		Delay (MCLK)
0	0	1
0	1	2
1	0	3
1	1	4

Bits 8-9 RAS# assertion to CAS# deassertion (RA2CD[0:1])

These two bits determine the duration between RAS# assertion to CAS# deassertion for CAS-before-RAS refresh cycle.

Bits 8-9		Delay (MCLK)
0	0	1
0	1	2
1	0	3
1	1	4

Bit 10-12 RAS# active to CAS# active delay

Bit 10-12			RAS# to CAS# delay (MCLK)
0	0	0	1
0	0	1	2
0	1	0	3
0	1	1	4
1	0	0	5
1	0	1	6
1	1	0	7
1	1	1	8

Bits 13-15 Last data in to Active command period during write cycle.

Bit 10-12			Last data-in to ACT period (MCLK)
0	0	0	1
0	0	1	2
0	1	0	3
0	1	1	4
1	0	0	5
1	0	1	6
1	1	0	7
1	1	1	8

DRAM Timing Control Register_2 (DRAMctrl2)

Index : 0x3e Read/Write Power-on Default : 0x0

0	1	2	3	4	5	6	7
LDI2PRE[0:1]		MCLK freq. select		ENECP	TEST	Mclken[0:1]	

8	9	10	11	12	13	14	15
Mclken[2]	MCLK_out_sel[0:2]			PRE_all			

Bits 0-1 Last data-in to precharge command during write cycle

Bit 0-1	CAS# active-time (MCLK)
0 0	1
0 1	2
1 0	3
1 1	4

Bits 2-3 MCLK frequency select

Bit 8-9	MCLK freq. select
0 0	CPUCLK
0 1	CPUCLK/1.5
1 0	CPUCLK/2
1 1	Reserved

Bit 4 ECP enable

When this bit is set, the GPIO[0:12] and GPIO[15:18] are redefined as parallel port interface. (refer to Table 5.3.1)

Bit 5 Test mode for outputing CPUCLK, MCLK_CTL and MCLK_DATA.

When this bit is set, as well as bit-4 is reset, the above 3 internal clocks are showed on GPIO[10:12]. This mode is used to adjust phase skew of MCLKs.

Bits 6-8 MCLK output buffer control.

Bit 6-8	MCLK driving capability (mA)
0 0 0	16
0 0 1	8
0 1 0	8
0 1 1	4
1 0 0	8
1 0 1	4
1 1 0	4
1 1 1	output tri-state

Bits 9-11 MCLK output delay reference CLKTREE MCLKO.

Bit 9-11	MCLK ouput delay

			(xo02d2 delay)
0	0	0	0
0	0	1	2
0	1	0	4
0	1	1	6
1	0	0	8
1	0	1	10
1	1	0	12
1	1	1	14

Bit 12 Always precharge all enable.
This bit set SDRAM state machine will always through precharge state and stop in idle state after each SDRAM read/write access.

Bits 4-15 Reserved

6.2 VIDEO ACCELERATOR REGISTERS

Video accelerator (VA) is consisted of video pre- (VPRE) and post-engine (VPOST). The VPRE handls the functions about alternative video inputs, like MPEG chips or camera. VPOST handles all the functions about pictures displaying to TV (interlace) or monitor (non-interlace). There are totally 33 registers using to set up all functions of VA, 28s are in VPOST with IO space of 0xf0000100 to oxf000019c, and 5s are in VPRE with IO space of 0xf00001c0 to 0xf00001dc.

6.2.1 VPOST REGISTERS

There are twenty-eight registers, with IO space allocated from 0xf0000100 to 0xf000019c , included in the VPOST.

Table : VPOST Register Map (IO base (BA) : 0xf0000000)

Port Addr.	Symbol	Access	Description
BA + 0x100	VPC	R/W	VPOST Control Register
BA + 0x104	OPWFSR	R/W	Background Stream Fetch Stop/Restart for Opaque

BA + 0x108	WKSEX	R/W	Window Key Start/End X Register
BA + 0x10c	WKSEY	R/W	Window Key Start/End Y Register
BA + 0x110	HWCSWX	R/W	HardWare Cursor Start/Width X Register
BA + 0x114	HWCSHY	R/W	HardWare Cursor Start/Height Y Register
BA + 0x118	GFXSCKM	R/W	Graphics Stream Color Key Mask Register
BA+ 0x11c	GFXSCK	R/W	Graphics Stream Color Key Register
BA + 0x120	OVLC	R/W	Overlay Control Register
BA + 0x124	GFXSSA	R/W	Graphics Stream Start Address Register
BA + 0x128	VASSA	R/W	VA Stream Start Address Register
BA + 0x12c	HWCSSA	R/W	H/W Cursor Stream Start Address Register
BA + 0x130	GFXVASS	R/W	Graphics/VA Stream Stride Register
BA + 0x134	HWCSS	R/W	H/W Cursor Stream Stride Register
BA + 0x138	GFXVASFF	R/W	Graphics/VA Stream Fetch Finish Register
BA + 0x13c	VASC	R/W	VA Scaling Control Register
BA + 0x140	LUTINDEX	R/W	Look-up-table Index Register
BA + 0x144	LUTDATA	R/W	Look-up-table Data Register
BA + 0x148	FF12T	R/W	FIFO 1/2 Threshold Register
BA + 0x14c	FF34T	R/W	FIFO 3/4 Threshold Register
BA + 0x150	VAYCADJ	R/W	VA Brightness/Contrast/HUE/Saturation Adjustment
BA + 0x154	SCF	R/W	Subcarrier Frequency Register
BA + 0x158	SCFIP	R/W	Subcarrier Frequency Initial Phase Register
BA + 0x15c	HTDEE	R/W	Horizontal Total/Display Enable End Register
BA + 0x160	HSYNCSE	R/W	HSYNC Start/End Register
BA + 0x164	VTDEE	R/W	Vertical Total/Display Enable End Register
BA + 0x168	VRSE	R/W	Vertical Retrace Start/End Register
BA + 0x16c	HRS	R/W	Horizontal Retrace Start Register
BA + 0x170	HWCBC	R/W	Hardware Cursor Background Color Register
BA + 0x174	HWCFC	R/W	Hardware Cursor Foreground Color Register
BA + 0x178	TVTWH	R/W	TV Encoder Test Width/Height Register
BA + 0x17c	VPTC	R/W	VPOST Test Control Register
BA + 0x180	FIFO1D	R/W	FIFO 1 Data Register
BA + 0x184	FIFO2D	R/W	FIFO 2 Data Register
BA + 0x188	FIFO3D	R/W	FIFO 3 Data Register
BA + 0x18c	FIFO4D	R/W	FIFO 4 Data Register
BA + 0x190	FIFO5D	R/W	FIFO 5 Data Register

BA + 0x194	DTORT	R	DTO ROM Test Register
BA + 0x198	VPTS	R	VPOST Test Status Register
BA + 0x19c	VPCTD	R	VPOST Counter Test Data Register

VPOST Control Register (VPC)

Port address : 0x00000100 Read/Write Power-on Default : 0x00000000

0	1	2	3	4	5	6	7
Reserved				Wait_Yoff			

8	9	10	11	12	13	14	15
Reserved	CSW	HP	VP	Fliker Mode		BPP	

16	17	18	19	20	21	22	23
BGSEL	GFX_HUP	GFX_VUP	OFFSET	AJEN	AJCTL	FAL_D	DIG_ON

24	25	26	27	28	29	30	31
DISP_ON	FLK_ON	GFX_EN	VA_EN	OPA_EN	HWC_EN	WINKEN	COLKEN

Bits 0-3 Reserved

Bits 4-7 Wait_Yoff

Wait_Yoff = 2 x SCLK/13.5MHz

Bit 8 Reserved

Bit 9 CbCr Swap for 8-bit YCbCr Video Output Mode

0 = Output sequence is Cb0, Y0, Cr0, Y1, Cb2, Y2, Cr2, Y3, etc. (CCIR-656)

1 = Output sequence is Cr0, Y0, Cb0, Y1, Cr2, Y2, Cb2, Y3, etc. (CCIR-656)

Bit 10 HSYNC Output Pin Polarity

0 = Negative sync pulse

1 = Positive sync pulse

Bit 11 VSYNC Output Pin Polarity

0 = Negative sync pulse

1 = Positive sync pulse

Bits 12-13 3-line Flicker-Free Filter Mode

00 = 2D filter with 121, 242, 121 weightings
01 = 1D filter with 121 weightings
10 = 2D filter with 131, 141, 131 weightings
11 = 1D filter with 565 weightings

Bits 14-15 BPP (Bits Per Pixel) for graphics stream
00 = 16-color mode
01 = 256-color mode
10 = 565 high color mode
11 = Reserved

Bit 16 Background Select
0 = Graphics
1 = VA video

Bit 17 Graphics Stream Horizontal 2x Up-Scaling (Replication)
0 = Disable
1 = Enable

Bit 18 Graphics Stream Vertical 2x Up-Scaling (Replication)
0 = Disable
1 = Enable

Bit 19 Odd/Even Field Data offset
0 = One line offset
1 = No offset

Bit 20 Brightness/Contrast/Hue/Saturation Adjustment Enable
0 = Disable
1 = Enable

Bit 21 Brghtness/Contrast/Hue/Saturation Adjustment Contal
0 = Full-screen adjustment by using VACADJ and VAYADJ registers
1 = Adjust data within windows defined by window keys only

Bit 22 P[0:7] Output Control
0 = Pixel data is sampled by rising-edge of PCLK(pixel clock) then output to P[0:7]
1 = Pixel data is sampled by falling-edge of PCLK then output to P[0:7]

Bit 23 Digital Video Output Enable
0 = Disable
1 = Enable

Bit 24 Screen On
1 = Screen On
0 = Screen Off

Bit 25 3-line Fliker-free Filter Enable
0 = Disable

1 = Enable

Bit 26 Graphics Stream Enable

0 = Disable

1 = Enable

Bit 27 VA Stream Enable

0 = Disable

1 = Enable

Bit 28 Window Opaque Enable

0 = Disable

1 = Enable

Bit 29 Hardware Cursor Enable

0 = Disable

1 = Enable

Bit 30 Window Key Enable

0 = Disable

1 = Enable

Bit 31 Color Key Enable

0 = Disable

1 = Enable

Backgground Stream Fetch Stop/Restart for Opaque Window Register (OPWFSR)

Port address : 0x00000104 Read/Write Power-on Default : 0x----

0	1	2	3	4	5	6	7
Reserved						OPWFR[0:1]	

8	9	10	11	12	13	14	15
OPWFR[2:9]							

16	17	18	19	20	21	22	23
Reserved						OPWFS[0:1]	

24	25	26	27	28	29	30	31
OPWFS[2:9]							

Bits 0-5 Reserved

Bits 6-15 Background Stream Fetch Restart for Opaque Window

A 10-bit value specifies the horizontal offset in DWORD mermory cycles the background stream is to restart fetching

Bits 16-21 Reserved

Bits 22-31 Background Stream Fetch Stop for Opaque Window

A 10-bit value specifies the horizontal offset in DWORD mermory cycles the background stream is to be hidden for opaque window display

Window Key Start/End X Register (WKSEX)

Port address : 0x00000108 Read/Write Power-on Default : 0x----

0	1	2	3	4	5	6	7
Reserved					Window Key Start X[0:2]		

8	9	10	11	12	13	14	15
Window Key Start X[3:10]							

16	17	18	19	20	21	22	23
Reserved					Window Key End X[0:2]		

24	25	26	27	28	29	30	31
Window Key End X[3:10]							

Bits 0-4 Reserved

Bits 5-15 Window Key Start X

A 11-bit value specifies the horizontal starting pixel position of the window

Bits 16-20 Reserved

Bits 21-31 Window Key End X

A 11-bit value specifies the last horizontal pixel position of the window

Window Key Start/End Y Register (WKSEY)

Port address : 0x0000010c Read/Write Power-on Default : 0x----

0	1	2	3	4	5	6	7
Reserved						WinKey_YS[0:1]	

8	9	10	11	12	13	14	15

Winkey_YS[2:9]

16	17	18	19	20	21	22	23
Reserved						WinKey_YE[0:1]	

24	25	26	27	28	29	30	31
WinKey_YE[2:9]							

Bits 0-5 Reserved

Bits 6-15 Window Key Start Y

A 10-bit value specifies the vertical starting scan line of the window

Bits 16-21 Reserved

Bits 22-31 Window Key End Y

A 10-bit value specifies the last vertical scan line of the window

HardWare Cursor Start/Width X Register (HWCSWX)

Port address : 0x00000110 Read/Write Power-on Default : 0x----

0	1	2	3	4	5	6	7
Reserved					Hardware Cursor Start X[0:2]		

8	9	10	11	12	13	14	15
Hardware Cursor Start X[3:10]							

16	17	18	19	20	21	22	23
Reserved							

24	25	26	27	28	29	30	31
Reserved	Hardware Cursor Width X						

Bits 0-4 Reserved

Bits 5-15 Hardware Cursor Start X

A 11-bit value specifies the horizontal starting pixel position of the H/W cursor

Bits 16-24 Reserved

Bits 25-31 Hardware Cursor Width X

A 7-bit value specifies the width of the hardware cursor, maxmum value = 64

HardWare Cursor Start/Height Y Register (HWCSHY)

Port address : 0x00000114 Read/Write Power-on Default : 0x----

0	1	2	3	4	5	6	7
						HW cursor start Y[1:0]	

8	9	10	11	12	13	14	15
Hardware Cursor Start Y[2:9]							

16	17	18	19	20	21	22	23
Reserved							

24	25	26	27	28	29	30	31
Reserved	Hardware Cursor Height Y						

Bits 0-4 Reserved

Bits 5-15 Hardware Cursor Start Y

A 10-bit value specifies the vertical starting scan line of the H/W cursor

Bits 16-24 Reserved

Bits 25-31 Hardware Cursor Height Y

A 7-bit value specifies the height of the hardware cursor, maxmum value = 64

Graphics Stream Color Key Mask Register (GFXCKM)

Port address : 0x00000118 Read/Write Power-on Default : 0x----

0	1	2	3	4	5	6	7
Reserved							

8	9	10	11	12	13	14	15
Reserved						Color Key Mask[0:1]	

16	17	18	19	20	21	22	23

Color Key Mask [2:9]

24	25	26	27	28	29	30	31
Color Key Mask [10:17]							

Bits 0-13 Reserved
Bits 14-19 Graphics Stream Color Key Mask Blue
Bits 20-25 Graphics Stream Color Key Mask Green
Bits 26-31 Graphics Stream Color Key Mask Red

Graphics Stream Color Key Register (GFXCK)

Port address : 0x0000011c Read/Write Power-on Default : 0x----

0	1	2	3	4	5	6	7
Reserved							

8	9	10	11	12	13	14	15
Reserved						Color Key [0:1]	

16	17	18	19	20	21	22	23
Color Key [2:9]							

24	25	26	27	28	29	30	31
Color Key [10:17]							

Bits 0-13 Reserved
Bits 14-19 Graphics Stream Color Key Blue
Bits 20-25 Graphics Stream Color Key Green
Bits 26-31 Graphics Stream Color Key Red

Overlay Control Register (OVLC)

Port address : 0x00000120 Read/Write Power-on Default : 0x00000000

0	1	2	3	4	5	6	7

Reserved

8	9	10	11	12	13	14	15
Reserved							

16	17	18	19	20	21	22	23
Reserved							

24	25	26	27	28	29	30	31
Reserved				OC0	OC1	OC2	OC3

Bits 0-27 Reserved

Bits 28-31 Overlay Display Select

These four 1-bit registers are selected by color key and window key as described below

Color Key	Window Key	Overlay Control
0	0	OC0
0	1	OC1
1	0	OC2
1	1	OC3

Each 1-bit register, when selected, is used to control current overlaying output as described below.

Color Key	Overlay Output
0	Graphics Stream
1	VA Stream

Notes: Background on the screen, controlled by OC0 when all keys are inative, should be either graphics stream or VA stream as specified by VPOSTCR_16. Which mean that OC0 should be programmed to either 0 (when VPOSTCR_16 = 1) or 1 (when VPOSTCR_16 = 0).

Graphics Stream Start Address Register (GFXSSA)

Port address : 0x00000124 Read/Write Power-on Default : 0x00000000

0	1	2	3	4	5	6	7
Reserved		GFXSA[0:5]					

8	9	10	11	12	13	14	15
GFXSA[6:13]							

16	17	18	19	20	21	22	23
GFXSA[14:21]							

24	25	26	27	28	29	30	31
GFXSA[22:29]							

Bits 0-1 Reserved

Bits 2-31 Graphics Stream Start Address

A 30-bit value specifies the offset in DWORD boundary from the start of the frame buffer for graphics data stream.

VA Stream Start Address Register (VASSA)

Port address : 0x00000128 Read/Write Power-on Default : 0x00000000

0	1	2	3	4	5	6	7
Reserved		VASA[0:5]					

8	9	10	11	12	13	14	15
VASA[6:13]							

16	17	18	19	20	21	22	23
VASA[14:21]							

24	25	26	27	28	29	30	31
VASA[22:29]							

Bits 0-1 Reserved

Bits 2-31 VA Stream Start Address

A 30-bit value specifies the offset in DWORD boundary from the start of the frame buffer for VA video data stream.

H/W Cursor Stream Start Address Register (HWCSSA)

Port address : 0x0000012c Read/Write Power-on Default : 0x00000000

0	1	2	3	4	5	6	7

Reserved	HWCSA[0:5]

8	9	10	11	12	13	14	15
HWCSA[6:13]							

16	17	18	19	20	21	22	23
HWCSA[14:21]							

24	25	26	27	28	29	30	31
HWCSA[22:29]							

Bits 0-1 Reserved

Bits 2-31 Hardware Cursor Stream Start Address

A 30-bit value specifies the offset in DWORD boundary from the start of the frame buffer for hardware cursor data stream.

Graphics/VA Stream Stride Register (GFXVASS)

Port address : 0x00000130 Read/Write Power-on Default : 0x----

0	1	2	3	4	5	6	7
Reserved					GFXSS[0:2]		

8	9	10	11	12	13	14	15
GFXSS[3:10]							

16	17	18	19	20	21	22	23
Reserved					VASS[0:2]		

24	25	26	27	28	29	30	31
VASS[3:10]							

Bits 0-4 Reserved

Bits 5-15 Graphics Stream Stride

This register specifies the DWORD offset of vertically adjacent pixels in the graphics stream.

Bit 16-20 Reserved

Bit 21-31 VA Stream Stride

The register specifies the DWORD offset of vertically adjacent pixel in the VA stream buffer.

H/W Cursor Stream Stride Register (HWCSS)

Port address : 0x00000134 Read/Write Power-on Default : 0x----

0	1	2	3	4	5	6	7

8	9	10	11	12	13	14	15

16	17	18	19	20	21	22	23
Reserved					HWCSS[0:2]		

24	25	26	27	28	29	30	31
HWCSS[3:10]							

Bits 0-20 Reserved

Bit 21-31 Hardware Cursor Stream Stride

The register specifies the DWORD offset of vertically adjacent pixel in the Hardware cursor stream buffer.

Graphics/VA Stream Fetch Finish Register (GFXVASFF)

Port address : 0x00000138 Read/Write Power-on Default : 0x----

0	1	2	3	4	5	6	7
Reserved						GFXSFF[0:1]	

8	9	10	11	12	13	14	15
GFXSFF[2:9]							

16	17	18	19	20	21	22	23
Reserved						VASFF[0:1]	

24	25	26	27	28	29	30	31

VASFF[2:9]

Bits 0-5 Reserved

Bits 6-15 Graphics Stream Fetch Finish

This register specifies the number of DWORD DRAM access cycle for a horizontal scan line fetching of graphics data stream

Bits 16-21 Reserved

Bits 22-31 VA Stream Fetch Finish

This register specifies the number of DWORD DRAM access cycle for a horizontal scan line fetching of VA video data stream

VA Scaling Control Register (VASC)

Port address : 0x0000013c Read/Write Power-on Default : 0x----

0	1	2	3	4	5	6	7
VUPS	Reserved		VA Vertical Scaling Factor[0:4]				

8	9	10	11	12	13	14	15
VA Vertical Scaling Factor[5:12]							

16	17	18	19	20	21	22	23
HUPS	Reserved		VA Horizontal Scaling Factor[0:4]				

24	25	26	27	28	29	30	31
VA Horizontal Scaling Factor[5:12]							

Bit 0 VA Vertical Up-scaling Method

0 = Replication

1 = Interpolation

Bits 1-2 Reserved

Bits 3-15 VA Vertical Scaling Factor

This 13-bit value specifies the vertical scaling factor of 0.5 (1/2 down-scaling), and 1.0 ~ 7.999 (up-scaling). Bits 0-2 specify the integral part and bits 3-12 specify the decimal part of the scaling factor. 1/2 downing-scaling will be done when this value < 1 (bits 0-2 = 000 and bits 3-12 don care). Scaling is disabled when this value = 1.000 (bits 0-2 = 001 and bits 3-12 = 000H).

Bit 16 VA Horizontal Up-scaling Method

0 = Replication

1 = Interpolation

Bits 19-31 VA horizontal Scaling Factor

This 13-bit value specifies the horizontal scaling factor of 0.5 (1/2 down-scaling), and 1.0 ~ 7.999 (up-scaling). Bits 0-2 specify the integral part and bits 3-12 specify the decimal part of the scaling factor. 1/2 downing-scaling will be done when this value < 1 (bits 0-2 = 000 and bits 3-12 don care). Scaling is disabled when this value = 1.000 (bits 0-2 = 001 and bits 3-12 = 000H).

Look-up-table Index Register (LUTINDEX)

Port address : 0x00000140 Read/Write Power-on Default : 0x----

0	1	2	3	4	5	6	7
Reserved							

8	9	10	11	12	13	14	15
Reserved							

16	17	18	19	20	21	22	23
Reserved							R/W_

24	25	26	27	28	29	30	31
LUT Index							

Bits 0-22 Reserved

Bit 23 LUT Read/Write Mode

Bits 24-31 LUT Index

This index value determines which color LUT location will be accessed

Notes: Color LUT is used for color mapping between pixel value of graphics stream in pseudo modes (16 and 256-color modes) and the display color on the screen.

Look-up-table Data Register (LUTDATA)

Port address : 0x00000144 Read/Write Power-on Default : 0x----

0	1	2	3	4	5	6	7
Reserved							

8	9	10	11	12	13	14	15

Reserved	LUT Data [0:1]

16	17	18	19	20	21	22	23
LUT Data [2:9]							

24	25	26	27	28	29	30	31
LUT Data [10:17]							

Bits 0-13 Reserved

Bits 14-31 LUT Data

A 18-bit value specifies the LUT data. Bits 0-5 specify the blue data. Bits 6-11 specify the green data. Bits 12-17 specify the red data.

FIFO 1/2 Threshold Register (FF12T)

Port address : 0x00000148 Read/Write Power-on Default : 0x----

0	1	2	3	4	5	6	7
			SRM1_HT				

8	9	10	11	12	13	14	15
			SRM1_LT				

16	17	18	19	20	21	22	23
			SRM2_HT				

24	25	26	27	28	29	30	31
			SRM2_LT				

Bits 0-2 Reserved

Bits 3-7 First FIFO High Threshold

When frame Buffer FIFO is filled to the threshold, the FIFO is ready to release DRAM access to other pending requests. Initial value is 18H.

Bits 8-10 Reserved

Bits 11-15 First FIFO Low Threshold

When frame Buffer FIFO is fetched to this threshold by graphics controller, a request is generated tp the DRAM controller for DRAM access. Initial value is 0CH.

Bits 16-18 Reserved

Bits 19-23 2nd FIFO High Threshold

When frame Buffer FIFO is filled to the threshold, the FIFO is ready to release DRAM access to other pending requests. Initial value is 18H.

Bits 24-26 Reserved

Bits 27-31 2nd FIFO Low Threshold

When frame Buffer FIFO is fetched to this threshold by graphics controller, a request is generated tp the DRAM controller for DRAM access. Initial value is 0CH.

FIFO 3/4 Threshold Register (FF34T)

Port address : 0x0000014c Read/Write Power-on Default : 0x----

0	1	2	3	4	5	6	7
			SRM3_HT				

8	9	10	11	12	13	14	15
			SRM3_LT				

16	17	18	19	20	21	22	23
			SRM4_HT				

24	25	26	27	28	29	30	31
			SRM4_LT				

Bits 0-2 Reserved

Bits 3-7 3th FIFO High Threshold

When frame Buffer FIFO is filled to the threshold, the FIFO is ready to release DRAM access to other pending requests. Initial value is 18H.

Bits 8-10 Reserved

Bits 11-15 3th FIFO Low Threshold

When frame Buffer FIFO is fetched to this threshold by graphics controller, a request is generated tp the DRAM controller for DRAM access. Initial value is 0CH.

Bits 16-18 Reserved

Bits 19-23 4th FIFO High Threshold

When frame Buffer FIFO is filled to the threshold, the FIFO is ready to release DRAM access to other pending requests. Initial value is 18H.

Bits 24-26 Reserved

Bits 27-31 4th FIFO Low Threshold

When frame Buffer FIFO is fetched to this threshold by graphics controller, a request is generated tp the DRAM controller for DRAM access. Initial value is 0CH.

VA Stream Brightness/Contrast/HUE/Saturation Adjustment Register (VAYCADJ)

Port address : 0x00000150 Read/Write Power-on Default : 0x----

0	1	2	3	4	5	6	7
Reserved				Contrast[0:3]			

8	9	10	11	12	13	14	15
Brightness							

16	17	18	19	20	21	22	23
Saturation				Reserved	HUE[0:2]		

24	25	26	27	28	29	30	31
HUE[3:10]							

Bits 0-3 Reserved

Bits 4-7 VA Contrast Value

A 4-bit contrast adjustment value allows adjustments in contrast from 1/8 to 15/8, in increments of 1/8. Bit 4 specifies the integral part and bits 5-7 specify the decimal part of this value. Contrast adjustment is implemented by multiplying the Y data by this constant.

Bits 8-15 VA Brightness Value

An 8-bit 2 complement value allows adjustments in brightness from -128 to +127, in increments of 1. Brightness adjustment is implemented by adding or subtracting this constant from the Y data.

Bits 16-19 VA Saturation Value

A 4-bit saturation adjustment value allows adjustments in saturation from 1/8 to 15/8, in increments of 1/8. Bit 16 specifies the integral part and bits 17-19 specify the decimal part of this value. Saturation adjustment is implemented by multiplying both Cb and Cr by this constant.

Bit 20 Reserved

Bit 21-31 VA HUE Value

An 11-bit hue adjustment value allows adjustments in hue from 0 degree to 360 degree, in increments of 0.176 degree. Hue adjustment is implemented by

Cb1 = Cb cosA + Cr sinA

Cr1 = Cr cosA - Cb sinA

Subcarrier Frequency Register (SCF)

Port address : 0x00000154 Read/Write Power-on Default : 0x----

0	1	2	3	4	5	6	7
SCF[0:7]							

8	9	10	11	12	13	14	15
SCF[8:15]							

16	17	18	19	20	21	22	23
SCF[16:23]							

24	25	26	27	28	29	30	31
SCF[24:31]							

Bits 0-31 Subcarrier Frequency

A 32-bit value specifies the subcarrier frequency for TV by using the following equation:

SCF value = (fsc/fvoclk) 2^32

Subcarrier frequency is generated from the stable VOCLK (27 MHz) by an internal DDA (Digital Differential Accumulator).

Subcarrier Frequency Initial Phase Register (SCFIP)

Port address : 0x00000158 Read/Write Power-on Default : 0x----

0	1	2	3	4	5	6	7
Reserved							

8	9	10	11	12	13	14	15
Reserved							

16	17	18	19	20	21	22	23
Subcarrier Frequency Initial Phase [0:7]							

24	25	26	27	28	29	30	31
Subcarrier Frequency Initial Phase [8:15]							

Bits 0-15 Reserved

Bits 16-31 Subcarrier Frequency Initial Phase

This 16-bit register specifies the initial phase between the color subcarrier and sync signal

Horizontal Total/Display Enable End Register (HTDEE)

Port address : 0x0000015c Read/Write Power-on Default : 0x----

0	1	2	3	4	5	6	7
Reserved					Horizontal Total [0:2]		

8	9	10	11	12	13	14	15
Horizontal Total [3:10]							

16	17	18	19	20	21	22	23
Reserved					Horizontal Display End[0:2]		

24	25	26	27	28	29	30	31
Horizontal Display Enable End[3:10]							

Bits 0-4 Reserved

Bits 5-15 Horizontal Total

An 11-bit value specifies the total number of pixels in the horizontal scan line interval including the retrace time.

Bits 16-20 Reserved

Bits 21-31 Horizontal Display Enable End

An 11-bit value specifies the total number of displayed pixels for one scan line.

HSYNC Start/End Register (HSYNCSE)

Port address : 0x00000160 Read/Write Power-on Default : 0x----

0	1	2	3	4	5	6	7
Reserved					HSYNC Start [0:2]		

8	9	10	11	12	13	14	15
HSYNC Start [3:10]							

16	17	18	19	20	21	22	23
Reserved					HSYNC End[0:2]		

24	25	26	27	28	29	30	31
HSYNC End[3:10]							

Bits 0-4 Reserved

Bits 5-15 HSYNC Start

An 11-bit value, programmed in pixels, at which the HSYNC signal becomes active.

Bits 16-20 Reserved

Bits 21-31 HSYNC End

An 11-bit value, programmed in pixels, at which the HSYNC signal becomes inactive.

Vertical Total/Display Enable End Register (VTDEE)

Port address : 0x00000164 Read/Write Power-on Default : 0x----

0	1	2	3	4	5	6	7
Reserved						Vertical Total [0:1]	

8	9	10	11	12	13	14	15
Vertical Total [2:9]							

16	17	18	19	20	21	22	23
Reserved						VDEE [0:1]	

24	25	26	27	28	29	30	31
VDEE [2:9]							

Bits 0-5 Reserved

Bits 6-15 Vertical Total

A 10-bit value specifies the total number of scan lines for one field on the screen, including the retrace time.

Bits 16-21 Reserved

Bits 22-31 Vertical Display Enable End

A 10-bit value specifies the total number of displayed scan lines for one field on the screen.

Vertical Retrace Start/End Register (VRSE)

Port address : 0x00000168 Read/Write Power-on Default : 0x----

0	1	2	3	4	5	6	7
Reserved						VRS [0:1]	

8	9	10	11	12	13	14	15
VRS[2:9]							

16	17	18	19	20	21	22	23
Reserved						VRE [0:1]	

24	25	26	27	28	29	30	31
VRE [2:9]							

Bits 0-5 Reserved

Bits 6-15 Vertical Retrace Start

A 10-bit value, programmed in scan lines, at which the vertical retrace becomes active.

Bits 16-21 Reserved

Bits 22-31 Vertical Retrace End

A 10-bit value, programmed in scan line, at which the vertical retrace between inactive.

Horizontal Retrace Start Register (HRS)

Port address : 0x0000016c Read/Write Power-on Default : 0x----

0	1	2	3	4	5	6	7
Reserved							

8	9	10	11	12	13	14	15
Reserved							

16	17	18	19	20	21	22	23
Reserved					Horizontal Retrace Start [0:2]		

24	25	26	27	28	29	30	31

Horizontal Retrace Start [3:10]

Bits 0-20 Reserved

Bits 21-31 Horizontal Retrace Start

An 11-bit value, programmed in pixels, at which the internal horizontal retrace becomes active. The internal horizontal retrace pulse width is fixed to 16 pixel clock cycles.

Hardware Cursor Background Color Register (HWCBC)

Port address : 0x00000170 Read/Write Power-on Default : 0x----

0	1	2	3	4	5	6	7
Reserved							

8	9	10	11	12	13	14	15
Hardware Cursor Background Color [0:7]							

16	17	18	19	20	21	22	23
Hardware Cursor Background Color [8:15]							

24	25	26	27	28	29	30	31
Hardware Cursor Background Color [16:23]							

Bits 0-7 Reserved

Bits 8-31 Hardware Cursor Background Color

A 24-bit specify the background color for the hardware graphics cursor. Only RGB 8:8:8 color mode, bits 8-15 have the red value, bits 16-23 have the green value, and bits 24-31 have the blue value.

Hardware Cursor Foreground Color Register (HWCFC)

Port address : 0x00000174 Read/Write Power-on Default : 0x----

0	1	2	3	4	5	6	7
Reserved							

8	9	10	11	12	13	14	15
Hardware Cursor Foreground Color [0:7]							

16	17	18	19	20	21	22	23
Hardware Cursor Foreground Color [8:15]							

24	25	26	27	28	29	30	31
Hardware Cursor Foreground Color [16:23]							

Bits 0-7 Reserved

Bits 8-31 Hardware Cursor Foreground Color

A 24-bit specify the foreground color for the hardware graphics cursor. Only RGB 8:8:8 color mode, bits 8-15 have the red value, bits 16-23 have the green value, and bits 24-31 have the blue value.

TV Encoder Test Width/Height Register (TVTWH)

Port address : 0x00000178 Read/Write Power-on Default : 0x----

0	1	2	3	4	5	6	7
Reserved							

8	9	10	11	12	13	14	15
Reserved				Analog Video Output		TV System	

16	17	18	19	20	21	22	23
TV Encoder Horizontal Test Width							

24	25	26	27	28	29	30	31
TV Encoder Vertical Test Height							

Bits 0-11 Reserved

Bits 12-13 Analog Video Output Mode

0x = RGB out, TV-encoder is off

10 = Composite Video

11 = S-Video + Composite Video

Bits 14-15 TV System

00 = PAL-B, D, G, H, N

01 = PAL_M

10 = NTSC

11 = Reserved

Bits 15-23 TV Encoder Horizontal Test Width

An 8-bit value specifies the horizontal total for the TV scan line when the TV encoder horizontal test is enabled (VPTCR_29 = 1). This register is not used during normal operation.

Bits 24-31 TV Encoder Vertical Test Height

An 8-bit value specifies the vertical total for the TV screen when the TV encoder vertical test is enabled (VPTCR_28 = 1). This register is not used during normal operation.

VPOST Test Control Register (VPTC)

Port address : 0x0000017c Read/Write Power-on Default : 0x----

0	1	2	3	4	5	6	7
Reserved							

8	9	10	11	12	13	14	15
			ARM	PRESET	SELF	Signature Input Select	

16	17	18	19	20	21	22	23
Reserved	VTEST	Counter Select					

24	25	26	27	28	29	30	31
State-Machine Select			Test Mode	TV_V	TV_H	ColorBar	GrayLevel

Bits 0-10 Reserved

Bits 11 Arm Signature Analyzer

0 = Disable

1 = Start signature analyzer operation

Bits 12 Preset Signature to Seed Value

0 = Preset signature to seed value when analysis begins

1 = Do not preset

Bits 13 Signature Analyzer Self Test

0 = Disable

1 = Enable

Bits 14-15 Signature Analyzer Input Select

Bits 14-15	Signature Analyzer Input Data

00	Null (all_zeros)
01	CP/R data
10	Y/G data
11	C/B data

Bits 16 Reserved

Bits 17 Digital Video, PIN I/O Mode

0 = Video In

1 = Video Output

Bits 18-23 VPOST Counter Test Select

000000 = SRM1 x address counter

000001 = SRM2 x address counter

000010 = SRM3 x address counter

000011 = SRM4 x address counter

000100 = SRM5 x address counter

000101 = SRM1 y address counter

000110 = SRM2 y address counter

000111 = SRM3 y address counter

001000 = SRM4 y address counter

001001 = SRM5 y address counter

001010 = SRM1 and SRM2 opaque counter in MCLK

001011 = SRM3, SRM4 and SRM5 opaque counter in MCLK

001100 = VA scaling counter.

001101 = CRTC horizontal and vertical counter

001110 = TV horizontal and vertical counter

001111 = Opaque counter in VOCLK

Bits 24-26 SRM State Machine Select

000 = State Machine of SRM1

001 = State Machine of SRM2

010 = State Machine of SRM3

011 = State Machine of SRM4

100 = State Machine of SRM5

Bit 27 VPOST Test Enable

0 = Disable

1 = Enable

Bit 28 TV Encoder Vertical Test Enable

0 = Disable

1 = Enable

Bit 29 TV Encoder Horizontal Test Enable

0 = Disable

1 = Enable

Bit 30 Color Bar Test Enable

0 = Disable

1 = Enable

Bit 31 Gray Level Test Enable

0 = Disable

1 = Enable

FIFO 1 Data Register (FIFO1D)

Port address : 0x00000180 Read/Write Power-on Default : 0x----

0	1	2	3	4	5	6	7
FIFO1D[0:7]							

8	9	10	11	12	13	14	15
FIFO1D[8:15]							

16	17	18	19	20	21	22	23
FIFO1D[16:23]							

24	25	26	27	28	29	30	31
FIFO1D[24:31]							

Bits 0-31 FIFO 1 Data

A read or write access to the FIFO1D register will incremet the FIFO address.

Note: A write access to the VPOSTCR_27 register, and set to 1. Then FIFO address will be reset to 0.

FIFO 2 Data Register (FIFO2D)

Port address : 0x00000184 Read/Write Power-on Default : 0x----

0	1	2	3	4	5	6	7
FIFO2D[0:7]							

8	9	10	11	12	13	14	15

FIFO2D[8:15]

16	17	18	19	20	21	22	23
FIFO2D[16:23]							

24	25	26	27	28	29	30	31
FIFO2D[24:31]							

Bits 0-31 FIFO 2 Data

A read or write access to the FIFO2D register will auto incremet the FIFO address.

Note: A write access to the VPOSTCR_27 register, and set to 1. Then FIFO address will be reset to 0.

FIFO 3 Data Register (FIFO3D)

Port address : 0x00000183 Read/Write Power-on Default : 0x----

0	1	2	3	4	5	6	7
FIFO3D[0:7]							

8	9	10	11	12	13	14	15
FIFO3D[8:15]							

16	17	18	19	20	21	22	23
FIFO3D[16:23]							

24	25	26	27	28	29	30	31
FIFO3D[24:31]							

Bits 0-31 FIFO 3 Data

A read or write access to the FIFO3D register will auto incremet the FIFO address.

Note: A write access to the VPOSTCR_27 register, and set to 1. Then FIFO address will be reset to 0.

FIFO 4 Data Register (FIFO4D)

Port address : 0x0000018c Read/Write Power-on Default : 0x----

0	1	2	3	4	5	6	7
FIFO4D[0:7]							

8	9	10	11	12	13	14	15
FIFO4D[8:15]							

16	17	18	19	20	21	22	23
FIFO4D[16:23]							

24	25	26	27	28	29	30	31
FIFO4D[24:31]							

Bits 0-31 FIFO 4 Data

A read or write access to the FIFO4D register will auto incremet the FIFO address.

Note: A write access to the VPOSTCR_27 register, and set to 1. Then FIFO address will be reset to 0.

FIFO 5 Data Register (FIFO5D)

Port address : 0x00000190 Read/Write Power-on Default : 0x----

0	1	2	3	4	5	6	7
FIFO5D[0:7]							

8	9	10	11	12	13	14	15
FIFO5D[8:15]							

16	17	18	19	20	21	22	23
FIFO5D[16:23]							

24	25	26	27	28	29	30	31
FIFO5D[24:31]							

Bits 0-31 FIFO 5 Data

A read or write access to the FIFO5D register will auto incremet the FIFO address.

Note: A write access to the VPOSTCR_27 register, and set to 1. Then FIFO address will be reset to 0.

DTO ROM Test Register (DTORTR)

Port address : 0x00000194 Read-Only Power-on Default : 0x----

0	1	2	3	4	5	6	7
Reserved							

8	9	10	11	12	13	14	15
Reserved							

16	17	18	19	20	21	22	23
DTO ROM [0:7]							

24	25	26	27	28	29	30	31
DTO ROM [8:15]							

Bits 0-15 Reserved

Bits 16-31 DTO ROM Test

A read access to the DTORTR register will auto incremet the FIFO address.

Note: A write access to the VPOSTCR_27 register, and set to 1. Then FIFO address will be reset to 0.

VPOST Test Status Register (VPTS)

Port address : 0x00000198 Read-Only Power-on Default : 0x----

0	1	2	3	4	5	6	7
Reserved			VPOST State Machine Status				

8	9	10	11	12	13	14	15
Reserved						DEN	CRC-Busy

16	17	18	19	20	21	22	23
Signature Data [0:7]							

24	25	26	27	28	29	30	31
Signature Data [8:15]							

Bits 0-2 Reserved

Bits 3-7 VPOST State Machine Status
Bits 8-13 Reserved
Bit 14 Vertical Display Enable Status
0 = Inactive
1 = Active
Bit 15 Signature Analyzer Status
0 = Idle
1 = Busy
Bit 16-31 Signature Data

VPOST Counter Test Data Register (VPCTD)

Port address : 0x0000019c Read-Only Power-on Default : 0x----

0	1	2	3	4	5	6	7
		VPOST Counter Test Data [0:5]					

8	9	10	11	12	13	14	15
VPOST Counter Test Data [6:13]							

16	17	18	19	20	21	22	23
VPOST Counter Test Data [14:21]							

24	25	26	27	28	29	30	31
VPOST Counter Test Data [22:29]							

Bits 0-1 Reserved
Bits 2-31 VPOST Counter Test Data
This register contain data output of counter under testing when the VPOST counter test is enable (VPTCR_27 = 1). Counter under testing is determined by VPOST counter test select register (VPTCR_18-23).

6.2.2 VPRE REGISTERS

There are eight registers, with IO space allocated from 0xf00001c0 to 0xf00001dc, included in VPRE.

Table : VPRE Register Map (IO

base (BA) : 0xf0000000)

Port Addr.	Symbol	Access	Description
BA + 0x1c0	VCC	R/W	Video Capture Control Register
BA + 0x1c4	CWSEX	R/W	Cropping Window Start/End X Register
BA + 0x1c8	CWSEY	R/W	Cropping Window Start/End Y Register
BA + 0x1cc	CVHW	R/W	Captured Video Height/Width Register
BA + 0x1d0	CSA0	R/W	Capture Frame Buffer 0 Start Address Register
BA + 0x1d4	CSA1	R/W	Capture Frame Buffer 1 Start Address Register
BA + 0x1d8	CTM	R/W	Capture Test Mode Register
BA+ 0x1dc	CSTMD	R	Capture SRAM Test Mode Data Register
BA+0x1e0	VIM	R/W	Video Interrupt Mode

Video Capture Control Register (VCC)

Port address : 0x000001c0 Read/Write Power-on Default : 0x00000c18

0	1	2	3	4	5	6	7
Reserved			VCAP_HT[4:0]				

8	9	10	11	12	13	14	15
Reserved			VCAP_LT[4:0]				

16	17	18	19	20	21	22	23
Reserved					FLT_ON	PWOFF	CKF

24	25	26	27	28	29	30	31
DBE	DBS	Byte Swap		HSP	VSP	SKP	VCEN

Bits 0-2 Reserved

Bits 3-7 Video Capture FIFO High Threshold

When video capture FIFO is filled to this threshold, a request is generated to the DRAM controller for DRAM access. Initial value is 18H

Bits 8-10 Reserved

Bits 11-15 Video Capture FIFO Low Threshold

When video capture FIFO is fetched to this threshold by DRAM controller, the FIFO is ready to release DRAM access to other pending requests. Initial value is 0cH

Bits 16-20 Reserved

Bit 21 VPRE Filter

0 = Turn Off

1 = Turn On

Bit 22 Video In, Power Down Off

0 = Power On

1 = Power Off

Bits 23 VCLK Falling Edge Latch

0 = input video data and signals are latched by rising edge of VCLK

1 = input video data and signals are latched by falling edge of VCLK

Bits 24 Double Buffering Enable

0 = Buffer 0 active

1 = Buffer 1 active

Bits 25 Double Buffing Status (Read-Only)

0 = Buffer 0 active

1 = Buffer 1 active

Bits 26-27 Input Video Stream Format

	YUV Input Video Stream Format
Bits 26-27	8-Bit Mode
00	Y, U, Y, V, ...
01	U, Y, V, Y, ...
10	Y, V, Y, U, ...
11	V, Y, U, Y, ...

Bits 28 HS Input Pin Polarity

0 = Negative sync pulse

1 = Positive sync pulse

Bits 29 VS Input Pin Polarity

0 = Negative sync pulse

1 = Positive sync pulse

Bits 30 Skip Field (Interlaced) or Frame (Non-interlaced)

0 = Capture all received fields/frames video data

1 = Capture every other received fields/frames video data

Bits 31 Video Capture Enable

0 = Disable

1 = Enable

Cropping Window Start/End X Register (CWSEX)

Port address : 0x000001c4 Read/Write Power-on Default : 0x----

0	1	2	3	4	5	6	7
Reserved				Cropping Window Start X[11:8]			

8	9	10	11	12	13	14	15
Cropping Window Start X[7:0]							

16	17	18	19	20	21	22	23
Reserved				Cropping Window End X[11:8]			

24	25	26	27	28	29	30	31
Cropping Window End X[7:0]							

Bits 0-3 Reserved
Bits 4-15 Cropping Window Start X
A 12-bit value specifies the number of pixels between the inactive edge of HS and the first cropped video pixel
Bits 16-19 Reserved
Bits 20-31 Cropping Window End X
A 12-bit value specifies the number of pixels between the inactive edge of HS and the last cropped video pixel.

Cropping Window Start/End Y Register (CWSEY)

Port address : 0x000001c8 Read/Write Power-on Default : 0x----

0	1	2	3	4	5	6	7
Reserved					Cropping Window Start Y[10:8]		

8	9	10	11	12	13	14	15
Cropping Window Start Y[7:0]							

16	17	18	19	20	21	22	23
Reserved					Cropping Window End Y[10:8]		

24	25	26	27	28	29	30	31
Cropping Window End Y[7:0]							

Bits 0-3 Reserved

Bits 4-15 Cropping Window Start Y
A 11-bit value specifies the number of pixels between the inactive edge of VS and the first cropped video data line
Bits 16-19 Reserved
Bits 20-31 Cropping Window End Y
A 11bit value specifies the number of pixels between the inactive edge of VS and the last cropped video data line

Captured Video Height/Width Register (CVHW)

Port address : 0x000001cc Read/Write Power-on Default : 0x----

0	1	2	3	4	5	6	7
Reserved					Capture Video Height[10:8]		

8	9	10	11	12	13	14	15
Capture Video Height[7:0]							

16	17	18	19	20	21	22	23
Reserved					Capture Video Width[10:8]		

24	25	26	27	28	29	30	31
Capture Video Width[7:0]							

Bits 0-3 Reserved
Bits 4-15 Capture Video Height
An 11-bit value specifies the height in line of the captured video which is down-scaled (or not) from the cropped video. Down-scaling is automatically done bu an internal DDA (Digital Differential Accumulator)
Bits 16-19 Reserved
Bits 20-31 Capture Video Width
An 11-bit value specifies the width in pixel of the captured video which is down-scaled (or not) from the cropped video. Down-scaling is automatically done by an internal DDA (Digital Differential Accumulator)

Capture Frame Buffer 0 Start Address Register (CSA0)

Port address : 0x000001d0 Read/Write Power-on Default : 0x----

0	1	2	3	4	5	6	7
		CSA0[31:26]					

8	9	10	11	12	13	14	15

CSA0[25:18]

16	17	18	19	20	21	22	23
CSA0[17:10]							

24	25	26	27	28	29	30	31
CSA0[9:2]							

Bits 0-1 Reserved
Bits 2-31 Capture Frame Buffer 0 Start Address
A 30-bit value specifies the offset in DWORD boundary from the start of the frame buffer for frame buffer 0 of the captured video.

Capture Frame Buffer 1 Start Address Register (CSA1)

Port address : 0x000001d4 Read/Write Power-on Default : 0x----

0	1	2	3	4	5	6	7
		CSA1[31:26]					

8	9	10	11	12	13	14	15
CSA1[25:18]							

16	17	18	19	20	21	22	23
CSA1[17:10]							

24	25	26	27	28	29	30	31
CSA1[9:2]							

Bits 0-1 Reserved
Bits 2-31 Capture Frame Buffer 1 Start Address
A 30-bit value specifies the offset in DWORD boundary from the start of the frame buffer for frame buffer 1 of the captured video.

Capture Test Mode Register (CTM)

Port address : 0x000001d8 Read/Write Power-on Default : 0x----

0	1	2	3	4	5	6	7
Pstatus[0:3]				Reserved			

8	9	10	11	12	13	14	15
			TME	Counter Test Mode Select			

16	17	18	19	20	21	22	23
Counter Test Mode Data[15:8]							

24	25	26	27	28	29	30	31
Counter Test Mode Data[7:0]							

Bits 0-3 Status of power-on setting
This 4 bits memorize states of **MD[20~23]** during power-on interval.
Firmware reads these bits to know what kind of target board is operating.

Bits 4-10 Reserved

Bits 11 Capture Test Mode Enable
0 = Disable
1 = Enable

Bits 12-15 Capture Counter Test Mode Select
0001 = Cropping H_counter
0010 = Cropping V_counter
0100 = Horizontal down-scaling DDA
1000 = Vertical down-scaling DDA

Bits 16-31 Capture Counter Test Mode data (Read-Only)

Capture SRAM Test Mode Data Register (CSTMD)

Port address : 0x000001dc Read/Write Power-on Default : 0x----

0	1	2	3	4	5	6	7
Capture SRAM Test Mode Data[31:24]							

8	9	10	11	12	13	14	15
Capture SRAM Test Mode Data[23:16]							

16	17	18	19	20	21	22	23
Capture SRAM Test Mode Data[15:8]							

24	25	26	27	28	29	30	31
Capture SRAM Test Mode Data[7:0]							

Bits 0-31 Capture SRAM Test Mode Data

Video INTR Mode (VIM)

Port address : 0x000001e0 Read/Write Power-on Default : 0x0000

0	1	2	3	4	5	6	7
Reserved							

8	9	10	11	12	13	14	15
Reserved							

16	17	18	19	20	21	22	23
				FLT_ON	VSI_ON	VSO_ON	CAP_ON

24	25	26	27	28	29	30	31
				FLT_INT	VSI_INT	VSO_INT	CAP_INT

Bits 0-19 Reserved
Bit 20 VPRE Filter Compelete Interrput Mode
0 = Turn Off
1 = Turn On
Bit 21 Video In VSYNC Interrupt Mode
0 = Turn Off
1 = Turn On
Bit 22 Video Out VSYNC Interrupt Mode
0 = Turn Off
1 = Turn On
Bit 23 Capture Compelete Interrupt Mode
0 = Turn Off
1 = Turn On
Bits 24-27 Reserved
Bit 28 VPRE Filter Compelete Interrupt Occur
Bit 29 Video In VSYNC Interrupt Occur
Bit 30 Video Out VSYNC Interrupt Occur
Bit 31 CaptureCompelete Interrupt Occur

6.3 DMA REGISTERS

There are twelve registers included in two channels Direct Memory Access (DMA) controller. The IO address map is allocated from 0xf0000200 to 0xf000022c.

Table 6.3-1 : DMA Register Map
(IO base (BA) : 0xf0000000)

Port Addr.	Symbol	Access	Description
BA + 0x200	SAR0	R/W	Channel 0 Source Address Register
BA + 0x204	TAR0	R/W	Channel 0 Target Address Register
BA + 0x208	LETH0	R/W	Channel 0 Length Register
BA + 0x20c	MOD0	R/W	Channel 0 Mode Control Register
BA + 0x210	SAR1	R/W	Channel 1 Source Address Register
BA + 0x214	TAR1	R/W	Channel 1 Target Address Register
BA + 0x218	LETH1	R/W	Channel 1 Length Register
BA+ 0x21c	MOD1	R/W	Channel 1 Mode Control Register
BA + 0x220	DBA0	R/W	DMA IO Device 0 Bass Address
BA + 0x224	DBA1	R/W	DMA IO Device 1 Base Address
BA + 0x228	LCAR0	R	Channel 0 Length Counter
BA + 0x22c	LCAR1	R	Channel 1 Length Counter

Source Addrsee Register (SAR0 and SAR1)

Port address : 0xf0000200 Read/Write Power-on Default : 0x00000000
Port address : 0xf0000210 Read/Write Power-on Default : 0x00000000

0	1	2	3	4	5	6	7
Source Address Register byte 0							

8	9	10	11	12	13	14	15
Source Address Register byte 1							

16	17	18	19	20	21	22	23
Source Address Register byte 2							

24	24	26	27	28	29	30	31
Source Address Register byte 3							

Bit 0-31 Source address register(SAR)

Define DMA transfer source address. In memory to memory mode, the source address should be set in word boundary.

Target Address Register (TAR0 and TAR1)

Port address : 0xf0000204 Read/Write Power-on Default : 0x00000000
Port address : 0xf0000214 Read/Write Power-on Default : 0x00000000

0	1	2	3	4	5	6	7
Target Address Register byte 0							

8	9	10	11	12	13	14	15
Target Address Register byte 1							

16	17	18	19	20	21	22	23
Target Address Register byte 2							

24	24	26	27	28	29	30	31
Target Address Register byte 3							

Bit 0-31 DMA target address register(TAR)

Define target address. In memory to memory mode, the target address should be set in word boundary.

Length Register (LETH0 and LETH1)

Port address : 0xf0000208 Read/write Power-on Default : 0x00000000
Port address : 0xf0000218 Read/write Power-on Default : 0x00000000

0	1	2	3	4	5	6	7
Reserved							

8	9	10	11	12	12	14	15
Reserved							LEN0

16	17	18	19	20	21	22	23
LEN1-8							

24	24	26	27	28	29	30	31
LEN9-16							

Bit 0-14 Reserved

Bit 15-31 Transfer Length (LEN)

LEN 0-16 indicate DMA transfer length with max 128k-byte transferring. In memory to memory tranfser mode, the length must in word boundary, because of vounting by word in length counter.

Mode Control Register (MOD0 and MOD1)

Port address : 0xf000020c Read/Write Power-on Default : 0x0000000f

Port address : 0xf000020c Read/Write Power-on Default : 0x00000000

0	1	2	3	4	5	6	7
DMAen	Reserved	TCIen	ECPen	TC	M2M	DEM	IOtype0

8	9	10	11	12	13	14	15
IOtype1	TRtype		IOrec				

16	17	18	19	20	21	22	23	
Wstate					FIX	Reserved	Tout0	(MOD0)
Wsate					DACK0L	DACK1L	CS0L	(MOD1)

24	25	26	27	28	29	30	31	
Tout1-8								(MOD0)
CS1L	DACK1A	TO0	TO1	Reserved				(MOD1)

Bit 0 DMA enable(DMAen)

1 = DMA transfer enable.
0 = DMA transfer disable

Bit 1 Reserved

Set to 0

Bit 2 Terninal count interrupt enable(TCIen)

1 = enable terminal count interrupt

Once this bit is set, and TC is asserted, the DMAC will generate external interrupt to host.

Bit 3 Enable ECP as DMA device(ECPen)

1 = ECP is set as DMA device

Bit 4 Terminal count flag(TC)

1 = indicate the length counter reaches 0, and theTC asserted

Bit 5 Memory to memory transfer(M2M)

1 = DMA is set to memory to memory transfer

0 = DMA memory to memory transfer is disable

Bit 6 Demand mode or block mode select(DEM)

1 = DMA transfer between memory and IO is demand mode

0 = DMA transfer between memory and IO is block mode

This bit is valid only in the transfer between memory and IO.

Bit 7-8 DMA IO device type(IOtype)

00 = 8-bit type, length counter(LENC) counts by byte

01 = 16-bit type, length counter counts by half word

10 = 32-bit type, length counter countes by word

11 = undefined

Only 8-bit external IO device is supported.

Bit 9-10 DMA transfer type(TRtype)

00 = memory to memory transfer

01 = memory to IO transfer

1x = IO to memory transfer

Bit 11-15 DMA IO read/write command recovery time(IOrec)

This field define the recovery cycle between two read/write command.

Bit 16-20 DMA IO read/write command wait state(Wstate)

This field define the IO read/write command wait state.

In MOD0:

Bit21 DMA transfer fix mode(FIX)

1 = Set DMA transfer as rotate mode. In rotate mode, the DMA controller acknowledge channel 1 request right after channel 0 being served. The channel 1 and channel 0 are served by turns.

0 = DMA is set in fix mode. Channel 0 is the most privilege. The channel 1 will not get the service token, unless channel 0 release the request.

Bit22 Reserved

This bit should be set to 0.

Bit23-31 Ready timeout counter(Tout)

Set IO device assert NOT ready timeout cycle count. When IO read/write command is issued, and if the IO device inserts wait state by asserting IORDY, the ready timeout counter starts to count. If the counter reach the Tout before read/write command is completed, the timeout flag TO0 or TO1 is to be set.

In MOD1:

Bit 21 Set DACK0 low active(DACK0L)

1 = set DMA acknowledge signal DACK0 to low active

0 = set DMA acknowledge signal DACK0 to high active

Bit 22 Set DACK1 low active(DACK1L)

1 = set DMA acknowledge signal DACK1 to low active

0 = set DMA acknowledge signal DACK1 to high active

Bit 23 Set CS0 low active(CS0L)

1 = Set IO device chip select CS0 to low active

0 = Set IO device chip select CS0 to high active

Bit 24 Set CS1 low active(CS1L)

1 = Set IO device chip select CS1 to low active

0 = Set IO device chip select CS1 to high active

Bit 25 DACK1 active

1 = indicate DMA channel 1 acknoewledge DACK1 is active

Bit 26 Channel 0 time out(TO0)

1 = indicate channel 0 IORDY signal timeout

This bit is read ONLY.

Bit 27 Channel 0 time out(TO1)

1 = indicate channel 1 IORDY signal timeout

This bit is read ONLY.

Bit28-31 Reserved

DMA IO Device Bass Address (DBA0 and DBA1)

Port address : 0xf0000220 Read/write Power-on Default : 0xfffff000

Port address : 0xf0000224 Read/write Power-on Default : 0xfffff000

0	1	2	3	4	5	6	7
DBA0-7							

8	9	10	11	12	12	14	15
DBA8-15							

16	17	18	19	20	21	22	23
DBA16-19				Reserved			

24	24	26	27	28	29	30	31
Reserved							

Bit0-19 DMA IO device base address(DBA)

Define DMA device IO base. The base address should not conflict to internal mega cell base, and the bit0-3 should be always set to 0.

Bit 20-31 Reserved

Length Counter Register (LCAR0 and LCAR1)

Port address : 0xf0000228 Read only Power-on Default : ---
Port address : 0xf000022c Read only Power-on Default : ---

0	1	2	3	4	5	6	7
Reserved							

8	9	10	11	12	12	14	15
Reserved							LENC0

16	17	18	19	20	21	22	23
LENC1-8							

24	24	26	27	28	29	30	31
LENC9-16							

Bit 0-14 Reserved

Bit 15-31 Length counter indicates the remainder to be transfer. DMA transfered number = Length Register - LENC. TC is asserted by Length counter reaching 0. Reading Length Counter may not get the valid value if the channel is active, for the length may be in transition.

6.4 PCI BRIDGE INTERFACE REGISTERS

There are four 32 bits registers included in the PCI Bridge Interface controller. The IO address map is allocated from 0xf0000250 to 0xf000025c.

Table 6.4-1 : PCI Bridge Register Map (IO base (BA) : 0xf0000000)

Port Addr.	Symbol	Access	Description
BA + 0x250	REG0	R/W	Master 0 Latency Register
BA + 0x254	REG1	R/W	Master 1 Latency Register
BA + 0x258	REG2	R/W	Master 2 Latency Register
BA + 0x25c	REG3	R/W	Master 3 Latency Register

Master 0 Latency Register (REG0)

Port address : 0xf0000250 Read/Write Power-on Default : 0x000003ff

0	1	2	3	4	5	6	7
Reserved							

8	9	10	11	12	13	14	15
Reserved							

16	17	18	19	20	21	22	23
PERRen	SERRen	Reserved			REQ0_reg[0:2]		

24	25	26	27	28	29	30	31
REQ0_reg[3:10]							

Bits 0-15 Reserved

Bit 16 System Error Enable
0 = disable
1 = enable

Bit 17 Parity Error Enable
0 = disable
1 = enable

Bits 18-20 Reserved

Bits 21-31 Number of PCICLK count for Master Latenercy Adjustment
Latenercy Time = REQ0_reg[0:11] / PCICLK

Master 1 Latency Register (REG1)

Port address : 0xf0000254 Read/Write Power-on Default : 0x000003ff

0	1	2	3	4	5	6	7
Reserved							

8	9	10	11	12	13	14	15
Reserved							

16	17	18	19	20	21	22	23
Reserved			CPURST	FIX	REQ1_reg[0:2]		

24	25	26	27	28	29	30	31
REQ1_reg[3:10]							

Bits 0-18 Reserved

Bit 19 CPU Reset Signal
0 = disable
1 = generate CPU Reset Signal

Bit 20 Request Priority Select
0 = Rotate Priority
1 = Fix priority

Bits 21-31 Number of PCICLK count for Master Latenercy Adjustment
Latenercy Time = REQ1_reg[0:11] / PCICLK

Master 2 Latency Register (REG2)

Port address : 0xf0000258 Read/Write Power-on Default : 0x000003ff

0	1	2	3	4	5	6	7
Reserved							

8	9	10	11	12	13	14	15
Reserved							

16	17	18	19	20	21	22	23
Reserved	PCICLK_sel[0:3]				REQ2_reg[0:2]		

24	25	26	27	28	29	30	31
REQ2_reg[3:10]							

Bits 0-16 Reserved

Bits 17-20 PCICLK output delay reference EXTCLK_.

Bit 9-11				PCICLK ouput delay (ns)
0	0	0	0	0
0	0	0	1	1
0	0	1	0	2
0	0	1	1	3
0	1	0	0	4
0	1	0	1	5
0	1	1	0	6
0	1	1	1	7
1	0	0	0	8
1	0	0	1	10
1	0	1	0	12
1	0	1	1	14
1	1	0	0	16
1	1	0	1	18
1	1	1	0	20
1	1	1	1	25

Bits 21-31 Number of PCICLK count for Master Latenency Adjustment
Latenency Time = REQ2_reg[0:11] / PCICLK

Master 3 Latency Register (REG3)

Port address : 0xf000025c Read/Write Power-on Default : 0x000003ff

0	1	2	3	4	5	6	7
Reserved							

8	9	10	11	12	13	14	15
Reserved							

16	17	18	19	20	21	22	23
Reserved					REQ3_reg[0:2]		

24	25	26	27	28	29	30	31
REQ3_reg[3:10]							

Bits 0-20 Reserved

Bits 21-31 Number of PCICLK count for Master Latenency Adjustment
Latenency Time = REQ3_reg[0:11] / PCICLK

6.5 AIO BUS CONTROLLER

W90221 Provides a ISA-like bus for low speed devices such as ROM, Flash or other 8-bit/16-bit IOs. The bus shares the 32-bit data/addr bus, 4-bit comm/byte bus and the INTD_ signal of PCI bus. AIO supports only 8-bit memory (like ROM/Flash) and 8- or 16-bit IO of devices attached on the bus. Also, the bus provides up to 16Mbyte (24 bits address lines) addressing space for memory and 64K (16 bits address lines) for IOs. Two base registers, 8-bit base for memory and 16 bit for IO, point to the start address of AIO-memory and AIO-IO space individually. System space 0xf0000360 ~ 0xf0000367 is allocated for three 16-bit configuration registers of AIO bus controller.

Table 6.5-1 : AIO Register Map
(IO base (BA) : 0xf0000000)

Port Addr.	Symbol	Access	Description
BA + 0x360	CFG[0:15]	R/W	Configuration Register
BA + 0x362	AIOBASE[0:15]	R/W	AIO-IO Space Base Register
BA + 0x364	XMBASE[7:15]	R/W	AIO-ROM Space Base Register
BA + 0x366	-	-	Reserved

Configuration Register (CFG)

Port address : 0xf0000360 Read/Write Power-on Default : 0x0

0	1	2	3	4	5	6
IO_EN	IOext			ROMWext		

7	8	9	10	11	12	13	14	15
ROMRext			CMDset		CMDhold		CMDrec	

Bits 0 AIO IO-space enable

0 = IO space disable
1 = IO space enable
IO devices connected on AIO bus will only be enabled by turned this bit on.

Bit 1-3 IOR/IOW command wait state
These 3 bits define the wait states of IOR or IOW commands on AIO bus. The IOR or IOW commands will be active for "CFG[1:3] + 2" PCICLK cycles.

Bit 4-6 Memory-Write (Flash write) command wait state
These 3 bits define the wait states of memory-wirte command on AIO bus. The write command will be active for "CFG[4:6] + 2" PCICLK cycles.

Bit 7-9 Memory-Read (ROM/Flash read) command wait state
These 3 bits define the wait states of memory-read command on AIO bus. The read command will be active for "CFG[7:9] + 2" PCICLK cycles.

Bit 10-11 command set-up time
These 2 bits define the address-to-command set-up time of all command cycles as well as data-to-command set-up time of write cycles. The set-up time is "CFG[10:11] + 2" PCICLK cycles refering falling edge of RD_/WR_ signals.

Bit 12-13 command hold time
These 2 bits define the address/data-to-command hold time of write cycles. The hold time is "CFG[12:13] + 2" PCICLK refering to rising edge of WR_ signal. As for read cycles, the minimum data hold time requirement is 0.

Bit 14-15 Recovery time of consective ROM read commands
For Flash/ROM read cycles, the AIO supports only 32-bit access. The AIO controller will convert the memory word access into 4 consective byte accesses automatically, and the latency
between consective RD_ cycles will be "CFG[14:15] + 2" PCICLK cycles.

AIO IO Base Register (AIOBASE)

Port address : 0xf0000362 Read/Write Power-on Default : 0x0

0	1	2	3	4	5	6	7
AIOBSED[0:7]							

8	9	10	11	12	13	14	15

AIOBASE[8:15]

Bit 0-15 AIO IO space base address

This register define the starting address of AIO's IO space on 64K boundary. As the high halfword (16 bits) address lines of **PCI accesses** match the AIOBASE[0:15] , and if CFG[0] has been enabled, AIO controller responds DEVSEL_ and TRDY_ to PCI bridge and issue a IO access cycle to AIO IO devices.

AIO Memory Base Register (XMBASE)

Port address : 0xf0000364 Read/Write Power-on Default : 0x1ef

0	1	2	3	4	5	6	7
Reserved							RAWS

8	9	10	11	12	13	14	15
XMBASE[0:7]							

Bit 7 Always ROM cycle

Set this bit to logic one, all PCI accesses will be redirected to AIO memory (ROM/Flash) accesses. The bit is set after each cold start so that the chip's initialization (ROM access) will be redirected to AIO bus where the code ROM attached. Turn off this bit, once the other **PCI** or **AIO devices** need to be enabled.

Bit 8-15 AIO memory space base address

This register define the starting address of AIO's memory space on 16M boundary. As the high byte address lines of **PCI accesses** match the XMBASE[0:7], AIO controller responds DEVSEL_ and TRDY_ to PCI bridge and issue a memory access cycle to AIO memory devices.

6.6 PARALLEL PORT INTERFACE REGISTERS

There are eleven registers included in the Parallel Port Interface (PPI) controller. The IO address map is allocated from 0xf0000370 to 0xf000037f.

Table 6.6-1 : PPI Register Map
(IO base (BA) : 0xf0000000)

Port Addr.	Symbol	Access	Description
BA + 0x378	DL	R/W	Data Line Register
BA + 0x379	DSR	R	Device Status Register
BA + 0x37a	DCR	R/W	Device Control Register
BA + 0x37b	FSR	R	FIFO Status Register
BA + 0x37c	FCR	R/W	FIFO Control Register
BA + 0x37d	IER	R/W	Interrupt Enable Register
BA + 0x37e	IIR	R	Interrupt Identification Register
BA+ 0x37f	DR	R	Data Register
BA + 0x370	Dfifo	R/W	Data FIFO
BA + 0x374	CMD	R/W	Command Register
BA + 0x375	TOR	R/W	Time Out Register
BA + 0x376 ~ BA + 0x377	-	-	Reserve for PPI future extension

Data Line Register (DL)

Port address : 0xf0000378 Read/Write Power-on Default : --

0	1	2	3	4	5	6	7
8-bit Data Lines status							

Bits 0-7 Data Lines status

This is the standard parallel port data register. Writing to this register will drive data to the parallel port data lines. Reads to this register return the value on the data lines.

Device Status Register (DSR)

Port address : 0xf0000379 Read only Power-on Default : ---

0	1	2	3	4	5	6	7
BUSY#	nACK	PE	SEL	nFAULT	EMPTY	FULL	CMDtrue

Bits 0 Inverted version of Parallel Port Interface "BUSY" signal

Bit 1 Version of Parallel Port Interface "nACK" signal

Bit 2 Version of Parallel Port Interface "PError" signal

Bit 3 Version of Parallel Port Interface "Select" signal

Bit 4 Version of Parallel Port Interface "nFault" signal

Bit 5 Echo device data FIFO "empty" status
0 = device data FIFO is not empty
1 = device data FIFO is empty

Bit 6 Echo device data FIFO "full" status
0 = device data FIFO is not full
1 = device data FIFO is full

Bit 7 "Command" pended
0 = Command Register (CMD) contains no command code
1 = A command code is in CMD not been transfered yet

This read-only register reflects the inputs on the Parallel Port Interface and some of device data FIFO and Command Register status.

Device Control Register (DCR)

Port address : 0xf000037a Read/write Power-on Default : 0x0

0	1	2	3	4	5	6	7
Reserved		DOE	nAck_len	nSELIN#	nINIT	nAUFD#	nSTB#

Bit 2 Data bus output enable
0 = Data bus is drived by PPI for forward transferring
1 = Data bus is drived by peripheral device for reverse transferring

This bit has no effects during "peripheral emulation mode", "standard mode", "fast standard mode" and "PS2 mode".

Bit 3 nACK interrupt enable

0 = Data bus is drived by PPI for forward transferring

1 = Data bus is drived by peripheral device for reverse transferring

When this bit is set. A low-to-high transition will generate a interrupt request to CPU core.

Bit 4 Complement version of Parallel Port Interface "nSelectIn" signal

Bit 5 Version of Parallel Port Interface "nInit" signal

Bit 6 Complement version of Parallel Port Interface "nAutoFd" signal

Bit 7 Complement version of Parallel Port Interface "nStrobe" signal

This register directly controls several output signals as well as enabling some functions. The power-on default "0x0" makes {nSelectIn, nInit, nAutoFd, nStrobe} in {high, low, high, high} state, and 8-bit data bus in output enable mode which are suit for "standard mode" transferring.

FIFO Status Register (FSR)

Port address : 0xf000037b Read only Power-on Default : ---

0	1	2	3	4	5	6	7
Dfifo valid bytes					DA	SA	OV

Bits 0-4 Valid bytes in device data FIFO (Dfifo)

During forward transferring, these bits indicate that how many bytes in 16-byte Dfifo still not be transfered yet. While during reverse transferring, these bits shows the number of data bytes which received from parallel port interface and not be read by CPU core.

Bit 5 Dfifo data available

0 = Dfifo contains data bytes less than one "PWord"

1 = Dfifo contains at least one "PWord" of valid data.

Bit 6 Dfifo space available

0 = Dfifo contains empty locations less than one "PWord"

1 = Dfifo contains at least one "PWord" of empty locations.

Bit 7 Dfifo over/under run

0 = Dfifo is not yet over- or under-run

1 = Dfifo is already over- or under-run

Once this bit is set, it will keep on set state until Dfifo or the PPI is reset.

FIFO Control Register (FCR)

Port address : 0xf000037c Read/Write Power-on Default : 0x0

0	1	2	3	4	5	6	7
DMAen	FRST	DRST	PWord	MOD		RDTH	

Bits 0 DMA mode enable

A low-to-high transition of this bit will make PPI issue a DREQ to DMA controller. On receiving the corresponding DACK, PPI deasserts the DREQ.

This bit will be cleared by DMA terminal-count (TC) asserting or by a CPU write cycle with data-in[0] = 0.

Bit 1 Reset Dfifo

Writing a logical one to this bit will assert "Dfifo Reset" for one EXTCLK cycle. This bit will return to deasserted state automatically after "Dfifo Reset" is issued.

Bit 2 Reset Device

Writing a logical one to this bit will assert "Device Reset" for one EXTCLK cycle. This bit will return to deasserted state automatically after "Device Reset" is issued.

Bit 3 PWord size

0 = PWord is 8 bits (1 byte)

1 = PWord is 32 bits (4 bytes)

"PWord" defines the basic unit of Dfifo access during CPU cycle.

Bit 4-5 Device mode select

IER[1] and FCR[4:5] are used to choose device operation mode.

{IER[1], FCR[4:5]}	Device Operation Mode
1 x 0	Test Mode
1 x 1	Peripheral Emulation Mode
0 0 0	Standard Mode
0 0 1	PS2 Mode
0 1 0	Fast Standard Mode
0 1 1	ECP Mode

Bit 6-7 Dfifo Read Threshold

These two bits define the threshold level for triggering data-available interrupt (Irpt_RDA) of Dfifo during reverse transferring.

	Read Threshold level	
FCR[6:7]	PWord = 1 byte	PWord = 4 bytes
0 0	16 bytes	16 bytes
0 1	12 bytes	12 bytes
1 0	8 bytes	8 bytes
1 1	1 byte	4 bytes

Interrupt Enable Register (IER)

Port address : 0xf000037d Read/Write Power-on Default : 0x0

0	1	2	3	4	5	6	7
Reserved	PEMU	Tout_Ien	TC_Ien	Temp_Ien	Rda_Ien	nFault_Ien	LOOP

Bits 1 Peripheral Emulation Mode enable

0 = Device is not operating in "Peripheral Emulation Mode" or "Test Mode"

1 = Set device to "Peripheral Emulation Mode" or "Test Mode"

This bit along with FCR[4:5] are used to choose device operation mode.

Bit 2 Time-Out Interrupt (Irpt_TOUT) enable

0 = Mask Irpt_TOUT

1 = Enable Irpt_TOUT

Bit 3 DMA Terminal-Count Interrupt (Irpt_TC) enable

0 = Mask Irpt_TC

1 = Enable Irpt_TC

Bit 4 Dfifo Empty Interrupt (Irpt_TEMP) enable

0 = Mask Irpt_TEMP

1 = Enable Irpt_TEMP

Bit 5 Dfifo Read Threshold Interrupt (Irpt_RDA) enable

0 = Mask Irpt_RDA

1 = Enable Irpt_RDA

Bit 6 "nFault" Interrupt (Irpt_nFault) enable

0 = Mask Irpt_nFault

1 = Enable Irpt_nFault

Bit 7 Loop back enable

0 = Loop-back disable

1 = Loop-back enable

During Loop-Back mode, {nStrobe, nAutoFd, nInit, nSelectIn} will be fed to {nAck, Busy, PError, nFault} internally. This mode is used only for test issue.

Interrupt Identification Register (IIR)

Port address : 0xf000037e Read only Power-on Default : ---

0	1	2	3	4	5	6	7
Reserved		Irpt_Tout	Irpt_TC	Irpt_Temp	Irpt_RDA	Irpt_nFault	Irpt_nAck

Bits 2 Time-Out Interrupt flag

"Set" situation : If IER[2] is set, and "Time out" is occured during parallel port transferring.

"Reset" situation : Reset device, or CPU reads Time-Out Register (TOR) or Dfifo being accessed either by CPU or parallel port interface transferring.

Bits 3 DMA Terminal Count Interrupt flag

"Set" situation : If IER[3] is set, and TC is asserted by DMA controller once DMA

transfer is done.

"Reset" situation : TC is deasserted by DMA controller.

Bits 4 Dfifo Empty Interrupt flag

"Set" situation : If IER[4] is set, and Dfifo is empty during "**forward transferring**".

"Reset" situation : CPU write new data into Dfifo.

Bits 5 Dfifo Read Threshold Interrupt flag

"Set" situation : If IER[5] is set, and data bytes received by Dfifo are exceeded the threshold level (defined in FCR[4:5]) during "**reverse transferring**".

"Reset" situation : CPU read Dfifo such that data bytes in Dfifo are below the threshhold level.

Bits 6 "nFault" Interrupt flag

"Set" situation : If IER[6] is set, and a high-to-low transition is on "nFault" pin.

"Reset" situation : CPU read Device Status Register (DSR).

Bits 7 "nAck" Interrupt flag

"Set" situation : If DCR[3] is set, and a low-to-high transition is on "nAck" pin.

"Reset" situation : CPU read Device Status Register (DSR).

Data Register (DR)

Port address : 0xf000037f Read only Power-on Default : ---

0	1	2	3	4	5	6	7
8-bit Data of latched Lines status							

Bits 0-7 **Latched** Line status

The status of data lines of PPI will be latched into this register if a high-to-low transition is happened on "nAck" pin.

This register is added to support "Peripheral Emulation Mode" operation.

Device Data FIFO (Dfifo)

Port address : 0xf0000370 Read/write Power-on Default : ---

0	1	2	3	4	5	6	7
Dfifo MSB byte							

8	9	10	11	12	13	14	15

16	17	18	19	20	21	22	23

24	25	26	27	28	29	30	31
Dfifo LSB byte							

The device build-in a 16-byte data fifo to accelerate the transfer rate when using "Fast Standard Mode" or "ECP mode".

The Dfifo may be 1-byte or 4-byte accessed by CPU using "PWord" basis.

Command Register (CMD)

Port address : 0xf0000374 Read/write Power-on Default : ---

0	1	2	3	4	5	6	7
Pended Command Code							

Bits 0-7 Pended Command Code

Whenever a command code is written by CPU, a "command trasfer" will be induced immediately during "**ECP forward transferring**".
If CMD contains a command code not been transfered yet, a "command pended" status (CMDtrue) is echoed in DSR[7]. "Reset device" or "CPU read CMD" or the pended command code is finished transferring, the CMDtrue will also be cleared.

Time Out Register (TOR)

Port address : 0xf0000375 Read/write Power-on Default : ---

0	1	2	3	4	5	6	7
TOUTen	TOUTcmp						

Bits 0 Time Out Counter enable

0 = Disable Time Out counter

1 = Enable Time Out counter

Bits 1-7 Time Out counter (TOUTcnt[0:6]) comparsion value

If "TOUTen" is set, the "TOUTcnt[0:6] will be reset first and then start counting whenever a new PPI transfer cycle is initiated. On detecting TOUTcnt[0:6] is equal to TOUTcmp[1:7], a "Time Out" flag will be set which in turn trigger a interrupt request (Irpt_TOUT) if IER[2] (Toutlen) is also set at that time.

The tick of Time-Out Counter is about 9.175ms (OSC/(2**21) where OSC = 14.318MHz). The maximum duration that Time-Out Counter can cover is about 1.17 sec (2**7 ticks).

6.7 COM PORT INTERFACE REGISTERS

W90221 Provides 2 COM ports to interface external RS232 devices. COM0 allocates 0xf00003f8 ~ oxf00003ff as its IO-space, while COM1 allocates 0xf00002f8 ~ 0xf00002ff as its IO-space.

Table 6.7-1 : COM0 Register Map
(IO base (BA) : 0xf0000000)

Port Addr.	Symbol	Access	Description
BA + 0x3f8, DLAB = 0	RBR[0:7]	R	Receiver Buffer Register
BA + 0x3f8, DLAB = 0	THR[0:7]	W	Transmitter Holding Register
BA + 0x3f9, DLAB = 1	IER[3:7]	R/W	Interrupt Enable Register
BA + 0x3f8, DLAB = 1	DLL[0:7]	R/W	Divisor Latch Register (LS)

BA + 0x3f9, DLAB = 1	DLM[0:7]	R/W	Divisor Latch Register (MS)
BA + 0x3fa	IIR[0:7]	R	Interrupt Identification Register
BA + 0x3fa	FCR[0:7]	W	FIFO Control Register
BA+ 0x3fb	LCR[0:7]	R/W	Line Control Register
BA + 0x3fc	MCR[0:7]	R/W	Modem Control Register
BA + 0x3fd	LSR[0:7]	R	Line Status Register
BA + 0x3fe	MSR[0:7]	R	MODEM Status Register
BA + 0x3ff	TOR[0:7]	R/W	Time Out Register

Table 6.7-2 : COM1 Register Map
(IO base (BA) : 0xf0000000)

Port Addr.	Symbol	Access	Description
BA + 0x2f8, DLAB = 0	RBR[0:7]	R	Receiver Buffer Register
BA + 0x2f8, DLAB = 0	THR[0:7]	W	Transmitter Holding Register
BA + 0x2f9, DLAB = 1	IER[3:7]	R/W	Interrupt Enable Register
BA + 0x2f8, DLAB = 1	DLL[0:7]	R/W	Divisor Latch Register (LS)
BA + 0x2f9, DLAB = 1	DLM[0:7]	R/W	Divisor Latch Register (MS)
BA + 0x2fa	IIR[0:7]	R	Interrupt Identification Register
BA + 0x2fa	FCR[0:7]	W	FIFO Control Register
BA+ 0x2fb	LCR[0:7]	R/W	Line Control Register
BA + 0x2fc	MCR[0:7]	R/W	Modem Control Register
BA + 0x2fd	LSR[0:7]	R	Line Status Register
BA + 0x2fe	MSR[0:7]	R	MODEM Status Register
BA + 0x2ff	TOR[0:7]	R/W	Time Out Register

Receiver Buffer Register (RBR)

Port address : 0xf00003f8, DLAB=0 (COM0) Read only Power-on Default : --
0xf00002f8, DLAB=0 (COM1)

0	1	2	3	4	5	6	7
8-bit Receiver Data							

Bits 0-7 Receiver Data
Reading this register, COM port returns 8-bit data receiving from SDI pin.

Transmitter Holding Register (THR)

Port address : 0xf00002f8, DLAB=0 (COM0) Write only Power-on Default : --
0xf00002f8, DLAB=0 (COM1)

0	1	2	3	4	5	6	7
8-bit Transmit Data							

Bits 0-7 Transmit Data
Writing to this register, COM port will sent out the data through SDO pin (THR[7] first).

Interrupt Enable Register (IER)

Port address : 0xf00003f9, DLAB=0 (COM0) Read/Write Power-on Default : 0x0
0xf00002f9, DLAB=0 (COM1)

0	1	2	3	4	5	6	7
				MOS_len	RLS_len	THRE_len	RDA_len

Bit 4 MODEM Status Interrupt (Irpt_MOS) Enable
0 = Mask Irpt_MOS
1 = Enable Irpt_MOS

Bit 5 Receiver Line Status Interrupt (Irpt_RLS) Enable
0 = Mask Irpt_RLS

1 = Enable Irpt_RLS

Bits 6 Transmitter Holding Register Empty Interrupt (Irpt_THRE) Enable
0 = Mask Irpt_THRE
1 = Enable Irpt_THRE

Bits 7 Receiver Data Available Interrupt (Irpt_RDA) and Time-Out Interrupt (Irpt_TOUT) Enable
0 = Mask Irpt_RDA and Irpt_TOUT
1 = Enable Irpt_RDA and Irpt_TOUT

Divisor Latch (low byte) Register (DLL)

Port address : 0xf00003f8, DLAB=1 (COM0) Read/write Power-on Default : 0x0
0xf00002f8, DLAB=1 (COM1)

0	1	2	3	4	5	6	7
Baud Rate Divisor (Low Byte)							

Bit 0-7 Low byte of baud rate dvisor

Divisor Latch (high byte) Register (DLM)

Port address : 0xf00003f9, DLAB=1 (COM0) Read/write Power-on Default : 0x0
0xf00002f9, DLAB=1 (COM1)

0	1	2	3	4	5	6	7
Baud Rate Divisor (High Byte)							

Bit 0-7 High byte of baud rate dvisor

The 16-bit Divisor ({DLM, DLL}) is used to determine the COM port's baud rate. The equation is

Baud Rate = Frequency input / {16 * [Divisor +2]}

Interrupt Identification Register (IIR)

Port address : 0xf00003fa (COM0) Read only Power-on Default : ---

0xf00002fa (COM1)

0	1	2	3	4	5	6	7
FMENo	RTHo[0:1]		DMOD	IID[0:2]			NOI

Bits 0 Status of "FIFO Mode Enable"
This bit echos if "FIFO mode" is enable or not. Since "FIFO mode" is always enable, this bit always shows logical 1 when CPU reading this register.

Bit 1-2 Status of RX FIFO threshold level
These bits show current setting of receiver FIFO threshold level (RTH). The meaning of RTH is defined in the following FCR description.

Bit 3 DMA mode select
The DMA function is **not implemented** in this version. Reading IIR, the bit-3 is always 0.

Bit 4-6 Interrupt Identification bits
The IID[0:2] along with NOI indicate current interrupt request from COM port

Bit 7 No Interrupt (NOI) pended

Table 6.7-3 : Interrupt Control Functions

IIR[4:7]	Priority	Interrupt Type	Interrupt Source	Interrupt Reset control
- - - 1	--	None	None	--
0 1 1 0	Highest	Receiver Line Status (Irpt_RLS)	Overrun Error or Parity Error or Framing Error or Break Interrupt	Reading the LSR
0 1 0 0	Second	Received Data Available (Irpt_RDA)	Receiver FIFO thres-hold level is reached	Receiver FIFO drops below the threshold level
1 1 0 0	Second	Receiver FIFO Time-out (Irpt_TOUT)	Receiver FIFO is non-empty and no activities are occured in receiver FIFO during the TOR defined time duration	Reading the RBR
0 0 1 0	Third	Transmitter Hoding Register Empty (Irpt_THRE)	Transmitter Holding Register Empty	Reading the IIR (if source of interrupt is Irpt_THRE) or writing into the THR

0 0 0 0	Fourth	MODEM Status (Irpt_MOS)	CTS, DSR, DCD bits chang state or RI bit changes from high to low	Reading the MSR

FIFO Control Register (FCR)

Port address : 0xf00003fa (COM0) Write only Power-on Default : 0x1
0xf00002fa (COM1)

0	1	2	3	4	5	6	7
RTH[0:1]		Reserved		DMOD	TXRST	RXRST	FMEN

Bits 0-1 RX FIFO interrupt (Irpt_RDA) trigger level

FCR[0:1]		Irpt_RDA trigger level (bytes)
0	0	01
0	1	04
1	0	08
1	1	14

Bit 4 DMA mode select
The DMA function is **not implemented** in this version.

Bit 5 Reset TX FIFO
Seting this bit will generate 1 OSC cycle reset pulse to reset TX FIFO. The TX FIFO becomes empty (TX-pointer is cleared to 0) after such reset. This bit is returned to 0 automatically after the reset pulse is generated.

Bit 6 Reset RX FIFO
Seting this bit will generate 1 OSC cycle reset pulse to reset RX FIFO. The RX FIFO becomes empty (RX-pointer is cleared to 0) after such reset. This bit is returned to 0 automatically after the reset pulse is generated.

Bit 7 FIFO mode enable
The UART0 and UART1 are always operated on FIFO mode. Writing this bit has no effect while reading this bit always get logical one.

Line Control Register (LCR)

Port address : 0xf00003fb (COM0) Read/Write Power-on Default : 0x0
0xf00002fb (COM1)

0	1	2	3	4	5	6	7
DLAB	BREAK	SPAR	EPAR	PAR	STOP	WLEN	

Bits 0 Divisor Latch Access Bit

0 = "2F8/3F8" and "2F9/3F9" are used to access RBR, THR or IER.

1 = "2F8/3F8" and "2F9/3F9" are used to access Divisor Latch Registers (DLL, DLM).

Bit 1 Break Control Bit

When this bit is set to a logic 1, the serial data output (SOUT) is forced to the **Spacing State** (logic 0). This bit affects SOUT only and has no effect on the transmitter logic.

Bit 2 Stick Parity Enable

0 = Disable Stick Parity

1 = The parity bit is transmitted and checked as a logic 1 if bit-3=0 (odd parity), or as a logic 0 if bit-3=1 (even parity).

This bit has effects only when bit-4 (Parity Bit Enable) is set.

Bit 3 Even Parity Enable

0 = Odd number of logic 1s is transmitted or checked in the data word bits and parity bit.

1 = Even number of logic 1s is transmitted or checked in the data word bits and parity bit.

This bit has effects only when bit-4 (Parity Bit Enable) is set.

Bit 4 Parity Bit Enable

0 = Praity bit is not generated (transmit data) or checked (receive data) during transfer.

1 = Parity bit is generated of checked between the "last data word bit" and "stop bit" of the serial data.

Bit 5 Number of "Stop bit"

0 = One "stop bit" is generated in the transmitted data.

1 = **One and a half** "stop bit" is generated in the transmitted data when **5-bit** word length is selected.

Two "stop bit" is generated when **6-, 7-** and **8-bit** word length is selected.

Bits 6-7 Word Length Select

LCR[6:7]	Character length
0 0	5 bits

0	1	6 bits
1	0	7 bits
1	1	8 bits

Modem Control Register (MCR)

Port address : 0xf00003fc (COM0) Read/Write Power-on Default : 0x0
0xf00002fc (COM1)

0	1	2	3	4	5	6	7
Reserved			LOOP	OUT2#	OUT1#	RTS#	DTR#

Bits 3 Enable Loop-Back mode
0 = Disable
1 = When loop-back is enable, the following signals is connected internally.
SOUT connects to **SIN** internally and SOUT pin is fixed logic 1.
DTR# connects to **DSR#** internally and DTR# pin is fixed logic 1.
RTS# connects to **CTS#** internally and RTS# pin is fixed logic 1.
OUT1# connects to **RI#** internally and OUT1# pin is fixed logic 1.
OUT2# connects to **DCD#** internally and OUT2# pin is fixed logic 1.

Bit 4 Complement version of OUT2# (user-designated output) signal

Bit 5 Complement version of OUT1# (user-designated output) signal

Bit 6 Complement version of RTS# (Request-To-Send) signal

Bit 7 Complement version of DTR# (Data-Terminal-Ready) signal

Writing 0x00 to MCR **set** DTR#, RTS#, OUT1# and OUT2# to logic 1s, while
writing 0x0f to MCR **reset** DTR#, RTS#, OUT1# and OUT2# to logic 0s.

Line Status Register (LSR)

Port address : 0xf00003fd (COM0) Read only Power-on Default : ---
0xf00002fd (COM1)

0	1	2	3	4	5	6	7
Err_RCVR	TEMT	THRE	BI	FE	PE	OE	DR

Bits 0 RX FIFO Error

0 = RX FIFO works normally

1 = There is at least one parity error (PE), framing error (FE) or break indication (BI) in the FIFO. LSR[0] is cleared when CPU reads the LSR and if there are no subsequent errors in the RX FIFO.

Bit 1 Transmitter Empty

0 = Either Transmitter Holding Register (**THR** - TX FIFO) or Transmitter Shift Register (**TSR**) are not empty.

1 = Both THR and TSR are empty.

Bit 2 Transmitter Holding Register Empty

0 = THR is not empty.

1 = THR is empty.

The THRE bit is set when the last data word of TX FIFO is transferred to TSR. This bit is reset concurrently with the loading of the THR (or TX FIFO) by the CPU. This bit also causes the UART to issue an interrupt (Irpt_THRE) to the CPU when IER[6]=1.

Bit 3 Break Interrupt indicator

This bit is set to a logic 1 whenever the received data input is held in the "spacing state" (logic 0) for longer than a full word transmission time (that is, the total time of "start bit" + data bits + parity + stop bits).

Bit 4 Framing Error indicator

This bit is set to a logic 1 whenever the received character did not have a valid "stop bit" (that is, the stop bit following the last data bit or parity bit is detected as a logic 0).

Bit 5 Parity Error indicator

This bit is set to a logic 1 whenever the received character did not have a valid "parity bit".

Bit 6 Overrun Error indicator

An overrun error will occur only after the RX FIFO is full and the next character has been completely received in the shift register. The ccharacter in the shift register is overwritten, but it is not transferred to the RX FIFO. OE is indicated to the CPU as soon as it happens and is reset whenever the CPU reads the contents of the LSR.

Bit 7 RX FIFO Data Ready

0 = RX FIFO is empty

1 = RX FIFO contains at least 1 received data word.

LSR[3:5] (BI, FE, PE) is revealed to the CPU when its associated character is at the top of the RX FIFO. These three error indicators are reset whenever the CPU reads the contents of the LSR.

LSR[3:6] (BI, FE, PE, OE) are the error conditions that produce a "receiver line status interrupt" (Irpt_RLS) when IER[5]=1. Read LSR clear Irpt_RLS.

Writing LSR is a null operation (not suggested).

Modem Status Register (MSR)

Port address : 0xf00003fe (COM0) Read only Power-on Default : ---

0xf00002fe (COM1)

0	1	2	3	4	5	6	7
DCD#	RI#	DSR#	CTS#	DDCD	TERI	DDSR	DCTS

Bits 0 Complement version of **Data Carrier Detect** (DCD#) input

Bits 1 Complement version of **Ring Indicator** (RI#) input

Bits 2 Complement version of **Data Set Ready** (DSR#) input

Bits 3 Complement version of **Clear to Send** (CTS#) input

Bits 4 DCD# state change

This bit is set whenever DCD# input has changed state, and it will be reset if the CPU reads the MSR.

Bits 5 Tailing edge of RI

This bit is set whenever TI# input has changed from high to low, and it will be reset if the CPU reads the MSR.

Bits 6 DSR# state change

This bit is set whenever DSR# input has changed state, and it will be reset if the CPU reads the MSR.

Bits 7 CTS# state change

This bit is set whenever CTS# input has changed state, and it will be reset if the CPU reads the MSR.

Whenever either of MSR[4:7] is set to logic 1, a Modem Status Interrupt is generated if IER[4]=1. Writing LSR is a null operation (not suggested).

Time Out Register (TOR)

Port address : 0xf00003ff (COM0) Read/Write Power-on Default : 0x0
0xf00002ff (COM1)

0	1	2	3	4	5	6	7
TOUT_en	TOUT_cmp						

Bits 0 Time-Out (interrupt) enable
The feature of Receiver Time-Out (interrupt) is enable only when TOR[0] = IER[7] = 1.

Bits 1-7 Time-Out (interrupt) comparator

The Time-Out counter is reset and start counting (the counting clock = baud rate) whenever the RX FIFO receives a new data word. Once the content of Time-Out counter (TOUT_cnt) is equal to that of Time-Out comparator (TOUT_cmp), a Receiver Time-Out interrupt Irpt_ TOUT) is generated if TOR[0] = IER[7] = 1.

A new incoming data word or RX FIFO empty clears Irpt_TOUT.

6.8 SYNCHRONOUS SERIAL INTERFACE REGISTERS

There are five registers included in the Synchronous Serial Interface (SSI) controller. The IO address map is allocated from 0xf0000380 to 0xf000038a.

Table 6.8-1 : SSI Register Map
(IO base (BA) : 0xf0000000)

Port Addr.	Symbol	Access	Description
BA + 0x380	Dfifo	R/W	Data FIFO
BA + 0x384	CFGH	R/W	High Configuration Register
BA + 0x386	CFGL	R/W	Low Configuration Register
BA + 0x388	CTRL	R/W	Control Register
BA + 0x38a	STUS	R/W	Status Register

Data FIFO Register (Dfifo)

Port address : 0xf0000380 Read/Write Power-on Default : --

0	1	2	3	4	5	6	7
Dfifo MSB Byte							

8	9	10	11	12	13	14	15

16	17	18	19	20	21	22	23

24	25	26	27	28	29	30	31
Dfifo LSB byte							

The device build-in a 48x16 or 24x32 data fifo to accelerate the transfer rate.

The Dfifo may be 16-bit or 32-bit accessed by CPU, the type of reading, 16-bit or 32-bit depends on RX_FIFO type, and the type of writing, 16-bit or 32-bit depends on TX_FIFO type.

Bits 0-31

PCM data in/out, Whether the MSB bits are sign- or zero-extension depends on MEXT (CFGH[6]).

High Configuration Register (CFGH)

Port address : 0xf0000384 Read/Write Power-on Default : 0x0000

0	1	2	3	4	5	6	7
SSIEN	LOOP	MASTER	LFMOD	Reserved	FACT	MEXT	SLEN[0]

8	9	10	11	12	13	14	15
SLEN[1:4]				WPF[0:3]			

Bits 0 SSI Enable

0 = SSI disable

1 = SSI enable

Bit 1 Loop back enable

0 = disable

1 = enable

Bit 2 SYNC master mode enable

0 = SYNC is input.

1 = SYNC is output.

Bit 3 Long Framing Mode

0 = Short framing

The first data will be available on the next SCLK cycle as SYNC is active. In this mode,

SYNC width is 1 SCLK.

1 = Long framing

The first data will be available on the same SCLK cycle as SYNC is active. In this mode,

SYNC width is 1 SLEN.

Bit 4 Reserved

Bit 5 Frame active level

0 = active high

1 = active low

Bit 6 RX-FIFO MSB extension

0 = fill 0 in redundant MSBs of RX-FIFO

1 = non-implement

Bit 7-11 Serial word length

Word length = SLEN[0:4] + 1.

The word length supported by SSI is from 1 to 32 bits.

SLEN[0:4] configure TX/RX also.

if SLEN[0:4] <= 15, FIFO will be configured as 48x16.
if 15 < SLEN[0:4] <= 31, FIFO will be configured as 24x32.

Bit 12-15 Words per Frame

Words per frame = WPF[0:3] + 1. (max. 16 words/frame)

Low Configuration Register (CFGL)

Port address : 0xf0000386 Read/write Power-on Default : 0x0000

0	1	2	3	4	5	6	7
BPF[0:7]							

8	9	10	11	12	13	14	15
SCLKDIV[0:7]							

Bit 0-7 Number of bits per frame

Bits per frame = BPF[0:7] + 1. (max. 256 bits/frame)

Bit 8-15 Serial clock divider

On master mode,the SCLK is an output and its frequency is

SCLK frequency = EXTCLK / (2*(SCLKDIV+1))

Control Register (CTRL)

Port address : 0xf0000388 Read/Write Power-on Default : 0x0000

0	1	2	3	4	5	6	7
DVRST	TXRST	RXRST	RXTH[0:1]		TXTH[0:1]		IntRxen

8	9	10	11	12	13	14	15
IntTXen	IntERRen	Reserved					

Bits 0 Device reset

This is a self-clear bit, ie. set this bit to 1, it will be clear to 0 automatically after 1 EXTCLK.

When this bit is set, all registers will be set to its default value and the controller will be also

set to its initial states.

Bit 1 Reset TX-FIFO

This is a self-clear bit, ie. set this bit to 1, it will be clear to 0 automatically after 1 EXTCLK.

When this bit is set, The TX-FIFO pointer will be cleared to 0, the TX-FIFO is empty immediately.

Bit 2 Reset RX-FIFO

This is a self-clear bit, ie. set this bit to 1, it will be clear to 0 automatically after 1 EXTCLK.

When this bit is set, The RX-FIFO pointer will be cleared to 0, the RX-FIFO is empty immediately.

Bit 3-4 RX-FIFO threshold level

00 = RX-FIFO full
01 = 3/4 RX-FIFO
10 = 1/2 RX-FIFO
11 = RX-FIFO non-empty

Bit 5-6 TX-FIFO threshold level

00 = TX-FIFO empty
01 = 1/4 TX-FIFO
10 = 1/2 TX-FIFO
11 = TX-FIFO non-full

Bit 7 RX-FIFO interrupt enable

0 = disable
1 = enable

Bit 8 TX-FIFO interrupt enable

0 = disable
1 = enable

Bit 9 RX-FIFO overrun interrupt enable

0 = disable
1 = enable

Bit 10-15 Reserved

Status Register (STUS)

Port address : 0xf000038a Read/Write Power-on Default : --

0	1	2	3	4	5	6	7

RXDA	TXSA	RXERR	Reserved	IntRX

8	9	10	11	12	13	14	15
IntTX	INTRERR	Reserved					

Bits 0 RX-FIFO data available

0 = There is no valid data word in RX-FIFO.

1 = There is at least one valid data word in RX-FIFO.

Bit 1 TX-FIFO space available

0 = There is no space available in TX-FIFO.

1 = The TX-FIFO can still accept at least one data word.

Bit 2 RX-FIFO overrun

0 = The RX-FIFO works well.

1 = The RX-FIFO is already overrun.

Once the RX-FIFO is overrun, this bit will keep active until RX-FIFO is reset.

Bit 3-6 Reserved

Bit 7 RX-FIFO interrupt request

0 = No RX-FIFO interrupt request

1 = A RX-FIFO interrupt request is pending

Set = Valid data words in RX-FIFO exceeds the threshold level.

Reset = Valid data words in RX-FIFO drops below the threshold level.

Bit 8 TX-FIFO interrupt request

0 = No TX-FIFO interrupt request

1 = A TX-FIFO interrupt request is pending

Set = Valid data words in TX-FIFO drops below the threshold level.

Reset = Valid data words in TX-FIFO exceeds the threshold level

Bit 9 RX-FIFO overrun interrupt request

0 = No RX-FIFO overrun interrupt request

1 = A RX-FIFO overrun interrupt request is pending

Set = When RX-FIFO is overrun.

Reset = Reset RX-FIFO or reset device.

Bit 10-15 Reserved

6.9 TIMER REGISTERS

There are four registers included in the Timer. The IO address map is allocated from 0xf0000040 to 0xf0000043.

Table 6.9-1 Timer Register Map
(IO base (BA) : 0xf0000000)

Port Addr.	Symbol	Access	Description
BA + 0x40	TCR1	R/W	Timer Control Register 1
BA + 0x41	TICR1	R/W	Timer Initial Control Register 1
BA + 0x42	TCR2	R/W	Timer Control Register 2
BA + 0x43	TICR2	R/W	Timer Initial Control Register 2

Timer Control Register1 (TCR1)

Port address : 0x00000040 Read/Write Power-on Default : --

0	1	2	3 23	24 31
TI	CE	IE	Reserved	Pre-scale

Bit 0 Timer interrupt bit : The timer sets this bit to one to indicate that it has decremented to zero. this bit remain one until software sets it to zero.

Bit1 Counter Enable bit : Setting the CE bit to one causes the timer to begin decrementing Setting the CE bit to zero stops the timer.

Bit2 Interrupt Enable bit : When IE is set to one and the counter decrements to zero, the timer asserts its interrupt signal to interrupt CPU.

Bit24_31 Pre-Scalar : A pre-scalar value can be used to divide the input clock.

Timer Initial Control Register1 (TICR1)

Port address : 0xf0000041 Read/Write Power-on Default : --

0 7	8 31
reserved	Timer Initial Count

Bit8_31 : A 24-bit register for the initial counter value.

Timer Control Register2 (TCR2)

Port address : 0xf0000042 Read/Write Power-on Default : --

0	1	2	3 23	24 31
TI	CE	IE	Reserved	Pre-scale

Bit 0 Timer interrupt bit : The timer sets this bit to one to indicate that it has decremented to zero. this bit remain one until software sets it to zero.

Bit1 Counter Enable bit : Setting the CE bit to one causes the timer to begin decrementing Setting the CE bit to zero stops the timer.

Bit2 Interrupt Enable bit : When IE is set to one and the counter decrements to zero, the timer asserts its interrupt signal to interrupt CPU.

Bit24_31 Pre-Scalar : A pre-scalar value can be used to divide the input clock.

Timer Initial Control Register1 (TICR2)

Port address : 0xf0000043 Read/Write Power-on Default : --

0 7	8 31
reserved	Timer Initial Count

Bit8_31 : A 24-bit register for the initial counter value.

Two 24-bit decrementing timers will be implemented, corresponding to the TCR1, TICR1 and TCR2, TICR2 independently. When the timers interrupt enable bit is set to one and the counter decrements to zero, the timer will assert the associated interrupt signal. The interrupt signal will assert one of the 32 external interrupts defined by the EI bits in the control register. When a timer reaches zero, the timer hardware reloads the counter with the value from the timer initial count register and continues decrementing.

7 ELECTRICAL SPECIFICATIONS

7.1 Absolute Maximum Ratings

Ambient temperature ..... 0 °C ~ 70 °C
Storage temperature ..... -40 °C ~ 125°C
Voltage on any pin ..... Vss-0.5V ~ Vcc+0.5V
Power supply voltage 7V
Injection current (latch-up testing) 100mA
Operating power dissipation 30mA/MHz

7.2 DC Specifications

(Normal test conditions : VDD5V = 5.0V+/- 5%, VDDi/VDDp/VDDl = 3.3V+/- 5%, TA = 0 °C ~ 70 °C unless otherwise specified)

SYMBOL	PARAMETER	CONDITION	MIN	MAX	UNIT
VDD5V	Power Supply		4.75	5.25	V
VDDi/VDDp	Power Supply		3.14	3.46	V
V_{IL}	Input Low Voltage			0.8	V
V_{IH}	Input High Voltage		2.0		V
V_{OL}	Output Low Voltage	I_{OUT} = 2,4,8 mA (*2)		V_{SS}+ 0.4	V
V_{OH}	Output High Voltage	I_{OUT} = -1,2,4 mA (*2)	2.4		V
I_{CC}	Supply Current	F_{cpu} = 100MHz		300	mA
I_{IH}	Input High Current	V_{IN} = 2.4 V (*1)		10	µA
I_{IL}	Input Low Current	V_{IN} = 0.4 V (*1)	-10	10	µA
I_{IHP}	Input High Current (pull-up)	V_{IN} = 2.4 V (*3)	-45	-15	µA
I_{ILP}	Input Low Current (pull-up)	V_{IN} = 0.4 V (*3)	-10		µA

Note *1 : Inpt leakage current (I_{IL}, I_{IH}) include those bi-directional pins which are in "input" mode (output disable).

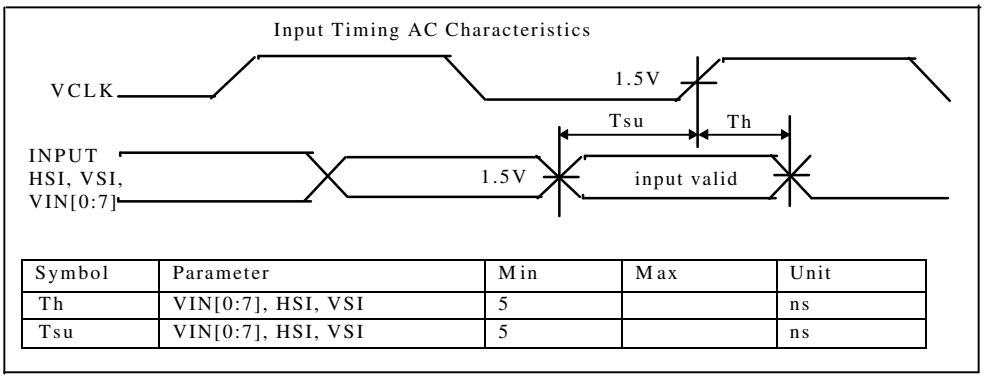
*2 : Pins of **4mA** sink capability include : DTR0n, RTS0n, SOUT0, SOUT0, HSO, VSO, SDO, VD[0:7],

SCLK, SYNC.
Pins of **6mA** sink capability include : PCIRST, PCICLK, GPIO[0:14].
Pins of **8mA** sink capability include : RAS#, CAS#, CKE, CS0H#, CS1H#, CS0L#, CS1L#, WE#,
DQMB[0:3], MA[0:13], GNT0#, GNT1#, INTD#, MD[0:31],
COMBE[0:3], PDA[0:31], X_STOP#, PERR#, FRAME#, IRDY#,
PPAR, TRDY#, DEVSEL#, GPIO[15:16].
Programmable 4/8/16mA sink capability : MCLK.
Current driver of full scale 18.7mA (analog output) : RED, GREEN, BLUE.
External Voltage reference pins : COMP, RSET, VREF.

*3 : Inputs with internal pull-up resistor include : GPIO[17:18], PREQ#[0:1], SERR#, INTA#, INTB#, INTC#,
CTS0n, DSR0n, RI0n, DCD0n, SDI.

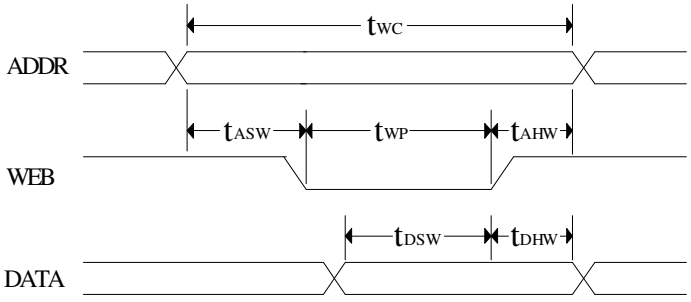
7.3 AC Specifications

Fig 7.3.1 Video-in bus timimg requirement



Symbol	Parameter	Min	Max	Unit
Th	VIN[0:7], HSI, VSI	5		ns
Tsu	VIN[0:7], HSI, VSI	5		ns

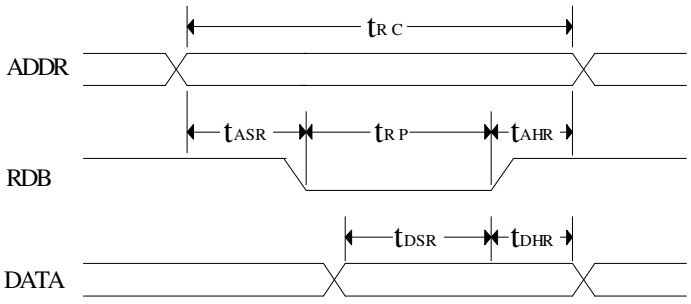
Fig 7.3.2 AIO- write timimg reqiorement



Symbol	Parameter	Min	Max	Unit
t_{wc}	Write cycle time	4		PCICLK
t_{asw}	ADDR to WEB setup time	1		PCICLK
t_{wp}	WEB pulse width	2		PCICLK
t_{ahw}	ADDR to WEB hold time	1		PCICLK
t_{dsw}	DATA to WEB setup time	2		PCICLK
t_{dhw}	DATA to WEB hold time	1		PCICLK

Fig 7.3.3 AIO- read timimg reqiorement

=



Symbol	Parameter	Min	Max	Unit
t_{rc}	Read cycle time	3		PCICLK
t_{asr}	ADDR to RDB setup time	1		PCICLK
t_{rp}	RDB pulse width	2		PCICLK
t_{ahr}	ADDR to RDB hold time	0		PCICLK
t_{dsr}	DATA to RDB setup time	2		PCICLK
t_{dhr}	DATA to RDB hold time	0		PCICLK

Fig 7.3.4 EDO-RAM timimg requirement

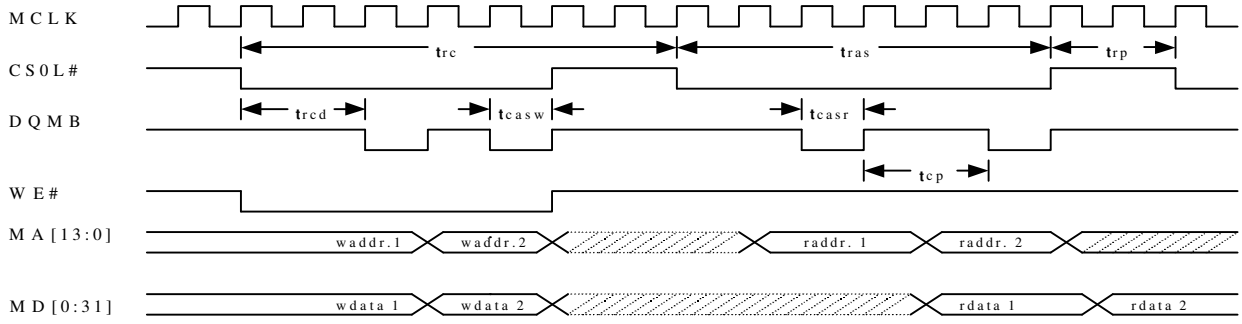
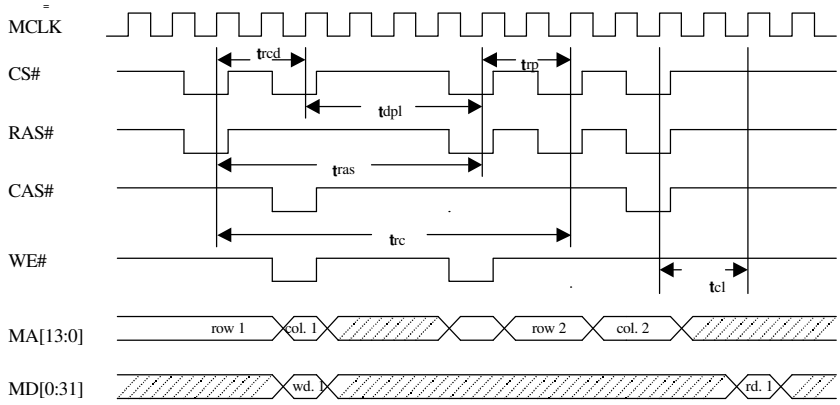


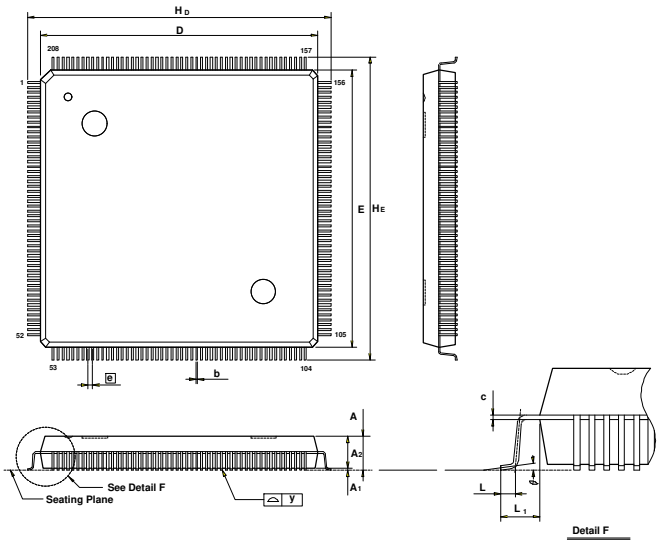
Fig 7.3.5 SDRAM timimg requirement



Symbol	Parameter	Min	Max	Unit
t_{rcd}	RAS# to CAS# delay	1	8	MCLK
t_{dpl}	Data-in to PRE Command Period	1	4	MCLK
t_{rp}	RAS# precharge time	1	8	MCLK
t_{ras}	RAS# active time	1	8	MCLK
t_{rc}	RAS# cycle time	2	9	MCLK
t_{cl}	CAS# latency time	1	3	MCLK
t_{casw}	CAS# pulse width for write	1	4	MCLK
t_{casr}	CAS# pulse width for read	1	8	MCLK
t_{cp}	CAS# precharge time	1	2	MCLK

8 PACKAGE DIMENSIONS

The W90220 is packaged in a 208-pin PQFP package. The following figure shows its mechanical dimension



Symbol	Dimension in inch			Dimension in mm		
	Min	Nom	Max	Min	Nom	Max
A	—	—	0.145	—	—	3.68
A_1	0.004	—	—	0.10	—	—
A_2	0.122	0.127	0.132	3.10	3.23	3.35
b	0.006	0.008	0.010	0.15	0.20	0.25
c	0.004	0.006	0.010	0.10	0.15	0.25
D	1.097	1.102	1.107	27.87	28.00	28.13
E	1.097	1.102	1.107	27.87	28.00	28.13
e	0.016	0.020	0.024	0.40	0.50	0.60
H_D	1.193	1.205	1.217	30.30	30.60	30.90
H_E	1.193	1.205	1.217	30.30	30.60	30.90
L	0.012	0.020	0.028	0.30	0.50	0.70
L_1	0.043	0.051	0.059	1.10	1.30	1.50
y	—	—	0.004	—	—	0.10
θ	0°	—	10°	0°	—	10°

Note:

1. Dimension D & E do not include interlead flash.
2. Dimension b does not include dambar protrusion/intrusion .
3. Controlling dimension : Millimeter
4. General appearance spec. should be based on final visual inspection spec.

APPENDIX A : ARCHITECTURE IMPLEMENT DEPENDENT REGISTERS

AIR[0] : Internal configuration register **(default : 12'b0)**
- bit 31 : Internal Icache enable (0/1 - disable/enable)
- bit 30 : Internal Dcache enable (0/1 - disable/enable)
- bit 29 : shall be the same as bit-30
- bit 28 : Default endian bit (0/1 - big-/little-endian)
- bit 27 : BTB enable (0/1 - disable/enable)
- bit 26 : reserved
- bit 25 : Multiplier fraction mode (0/1 - integer/fraction mode)
- bit 24 : reserved
- bit 23 : Freeze 1st 1K of Icache (0/1 - disable/enable)
- bit 22 : Freeze 2nd 1K of Icache (0/1 - disable/enable)
- bit 21 : Freeze 3rd 1K of Icache (0/1 - disable/enable)
- bit 20 : Freeze 4th 1K of Icache (0/1 - disable/enable)

AIR[1] : PSW register - for testing only

AIR[2] : TMR register - for testing only

AIR[3] : NonCacheable Offset register **(default : 32'b0)**
- bit 0-15 : reserve
- bit 16-19 : system non-cacheabel region
  - 0000 : all system memory are cacheable
  - 0001 : system memory above 1M are non-cacheable
  - 0010 : system memory above 2M are non-cacheable
  - 0011 : system memory above 4M are non-cacheable
  - 0100 : system memory above 8M are non-cacheable
  - 0101 : system memory above 16M are non-cacheable
  - 0110 : system memory above 32M are non-cacheable
  - 0111 : system memory above 64M are non-cacheable
  - 1000 : system memory above 128M are non-cacheable
  - 1001 : system memory above 256M are non-cacheable
- bit 20 : 0xA0000 ~ 0xFFFFF are non-cacheable
  - 0 : No
  - 1 : Yes
- bit 21-23 : Size of non-cacheable region 1
  - 000 : disable
  - 001 : 64K
  - 001 : 128K
  - 001 : 256K
  - 001 : 512K
  - 001 : 1M
  - 001 : 2M
  - 001 : 4M
- bit 24-26 : Size of non-cacheable region 2

The bit definition is the same as that of bit 21-23. The base address, defined by non-
cacheable base register (AIR[4]), will be on each regions' boundary automatically. (ie,
according to the size of each non-cacheable region, some LSBs of their relative base
register will be neglected.

- bit 27-29 : reserved
- bit 30 : Data cache write-through mode
  0 : write-back data cache
  1 : write-through data cache
- bit 31 : reserved

AIR[4] : NonCacheable Base register **(default : 32'b0)**
- bit 0-15 : base address of non-cacheable region 1
- bit 16-31 : base address of non-cacheable region 2

AIR[5] : reserved

AIR[6] : reserved

AIR[7] : (HPSW[0:31])
Back up states of PSW as pipeline enter HALT state.

AIR[8] : (ICEA_Front, ICEA_Back)
Back up IAOQ_Frond and IAOQ_Back as pipeline enter HALT state

AIR[9] : (IDR[0:31]) ICE-Data register
This register is used for data exchange between ICE module and CPU.

AIR[10] : (ITR[0:31]) ICE-Trap register
This register defines which trap event would force pipeline enter HALT state.

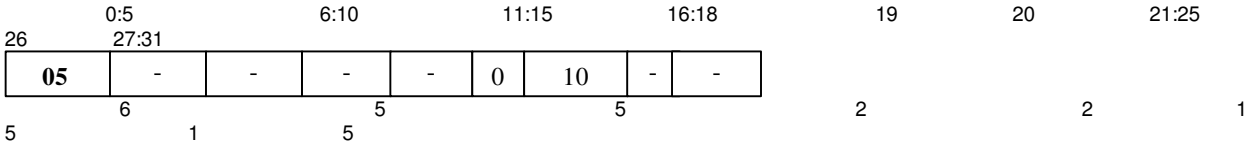
AIR[11] : (PWR[0:2]) Power-Mode register
- bit 0 : (SLEEP) Force CPU into sleep mode
- bit 1 : (DOZE) Force CPU to doze-mode
- bit 2 : (STDBY) Force CPU to stand-by-mode

AIR[12] : (CKR[0:7]) Stand-By-Clock register
Define the clock rate (STDCLK) in Stand-By mode :
STDCLK = CPUCLK /((CKR[0:7]+1)*4)

APPENDIX B : DIAGNOSTIC EXTENDED INSTRUCTION SET

EXIT

Format: EXIT



Purpose: Returning instruction pipeline to normal mode from ICE mode.

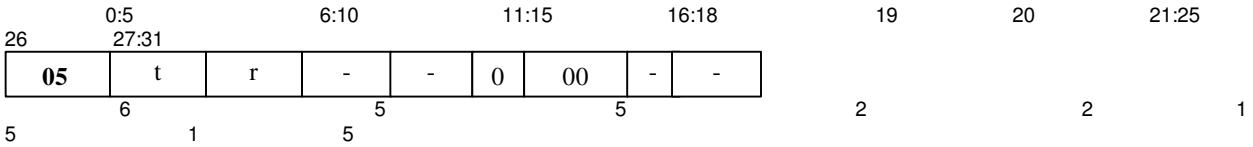
Description: Instruction pipeline will be back to normal mode immediately after "EXIT" has being executed.

Operation: PSW <-- HPSW;
IAOQ_Front <-- ICEA_Front;
IAOQ_Back <-- ICEA_Back;

Note : This instruction can be executed by code running at any privileged level, different from any other diagnostic instructions.

Move to AIR

Format: MTAIR r, t



Purpose : To copy value into a specified AIR from a general register.

Description : If the AIR[t] is existed, the contents of GR[r] is copied into AIR[t]. If AIR[t] has n bits where n <= 32, the least significant n bits of GR[r] are moved into AIR[t].

Operation : if (t > 3)
undefine operation;
else if (priv != 0)

```
        privilege instruction trap;
    else
        AIR[t] <-- GR[r];
```

Exception : Privilege instruction trap.

Restriction : This instruction can only be executed by code running at the most privilege level.

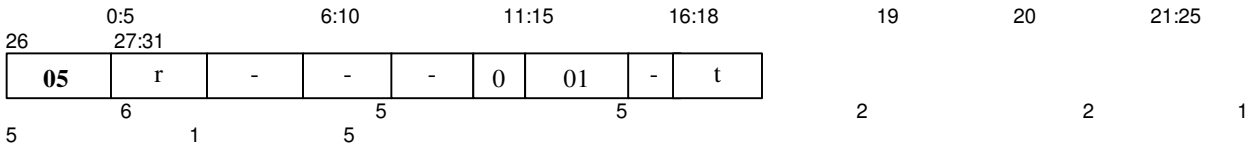
Note : AIR[0] : Internal configuration register **(default : 12'b0)**

- bit 31 : Internal Icache enable (0/1 - disable/enable)
- bit 30 : Internal Dcache enable (0/1 - disable/enable)
- bit 29 : shall be the same as bit-30
- bit 28 : Default endian bit (0/1 - big-/little-endian)
- bit 27 : BTB enable (0/1 - disable/enable)
- bit 26 : reserved
- bit 25 : Multiplier fraction mode (0/1 - integer/fraction mode)
- bit 24 : reserved
- bit 23 : Freeze 1st 1K of Icache (0/1 - disable/enable)
- bit 22 : Freeze 2nd 1K of Icache (0/1 - disable/enable)
- bit 21 : Freeze 3rd 1K of Icache (0/1 - disable/enable)
- bit 20 : Freeze 4th 1K of Icache (0/1 - disable/enable)

AIR[1] : PSW register **(default : 32'b0)**
AIR[2] : TMR register
AIR[3] : NonCacheable Offset register
AIR[4] : NonCacheable Mask register
AIR[5] : Write-Through Offset register
AIR[6] : Write-Through Mask register
AIR[7] : HPSW[0:31]
AIR[8] : ICEA_Front, ICEA_Back
AIR[9] : IDR[0:31] (ICE-Data register)
AIR[10] : ITR[0:31] (ICE-Trap register)
AIR[11] : PWR[0:2] (Power-Mode register)
AIR[12] : CKR[0:7] (Stand-By-Clock register)

Move from AIR

Format: MFAIR r, t



Purpose : To copy value into a general register from AIR register.

Description : If the AIR[t] is existed, the contents of AIR[r] is copied into GR[t]. If AIR[r] has n bits where n <= 32, the least significant n bits of AIR[r] are moved into GR[t].

Operation :
```
if (t > 3)
        undefine operation;
else if (priv != 0)
        privilege instruction trap;
else
        GR[t] <-- AIR[r];
```

Exception : Privilege instruction trap.

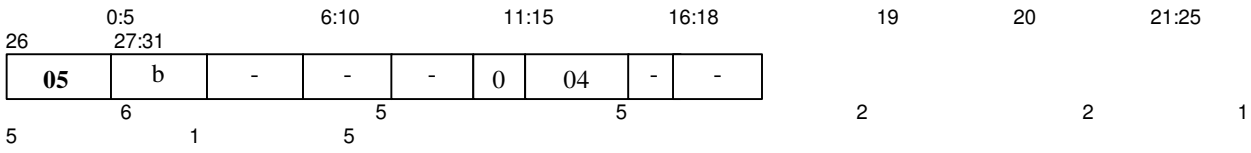
Restriction : This instruction can only be executed by code running at the most privilege level.

Note : AIR[0] : Internal configuration register **(default : 12'b0)**

AIR[1] : PSW register **(default : 32'b0)**
AIR[2] : TMR register
AIR[3] : NonCacheable Offset register
AIR[4] : NonCacheable Mask register
AIR[5] : Write-Through Offset register
AIR[6] : Write-Through Mask register
AIR[7] : HPSW[0:31]
AIR[8] : ICEA_Front, ICEA_Back
AIR[9] : IDR[0:31] (ICE-Data register)
AIR[10] : ITR[0:31] (ICE-Trap register)
AIR[11] : PWR[0:2] (Power-Mode register)
AIR[12] : CKR[0:7] (Stand-By-Clock register)

Move to Btag

Format: MTBTAG b



Purpose : To copy a value into BTB_tag from a general register.

Description : GR[b][0:23] is copied into a specified entry of BTB_tag.

Operation :
```
entry <-- GR[b][26:31];
set   <-- GR[b][24:25];
```

Btag[set, entry][0:26] <-- {GR[b][0:23], GR[b][21:23]};

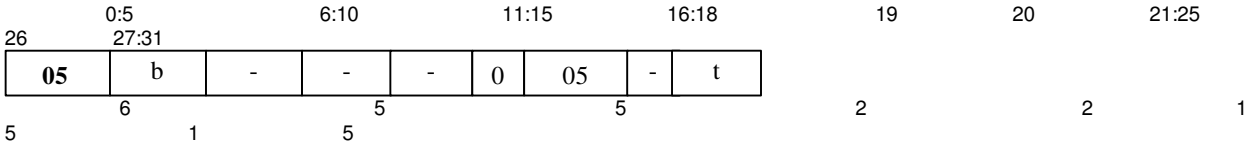
Exception : Privilege instruction trap.

Restriction : This instruction can only be executed by code running at the most privilege level.

Note : Btag[set, entry][0:23] : tag field
Btag[set, entry][24] : valid bit
Btag[set, entry][25:26] : LRU bits

Move from Btag

Format: MFBTAG b,t



Purpose : To copy a value into a general register from BTB_tag.

Description : A specified entry of BTB_tag is copied into GR[t].

Operation : entry <-- GR[b][26:31];
set <-- GR[b][24:25];

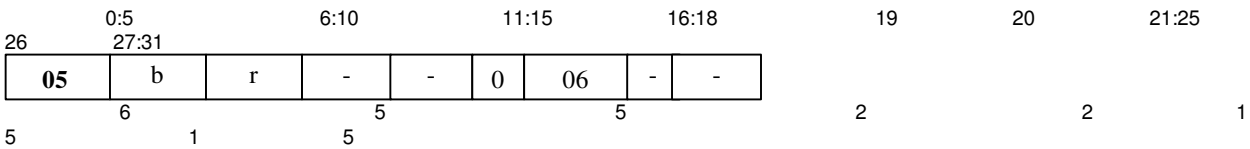
GR[t][0:26] <-- Btag[set, entry][0:26];

Exception : Privilege instruction trap.

Restriction : This instruction can only be executed by code running at the most privilege level.

Move to Btable

Format: MTBTAB r,b



Purpose : To copy a value into BTB_table from a general register.

Description : GR[r] is copied into a specified entry of BTB_table.

Operation :
```
entry <-- GR[b][26:31];
set       <-- GR[b][24:25];

Btable[set, entry][0:33] <-- {GR[r][0:31], GR[r][30:31]};
```

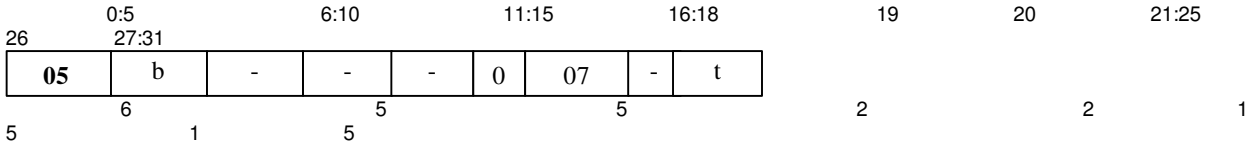
Exception : Privilege instruction trap.

Restriction : This instruction can only be executed by code running at the most privilege level.

Note : Btable[set, entry][0:31] : Branch target
Btable[set, entry][32:33] : History bits

Move from Btable

Format: MFBTAB b,t



Purpose : To copy a value into a general register from BTB_table.

Description : A specified entry of BTB_table is copied into GR[t].

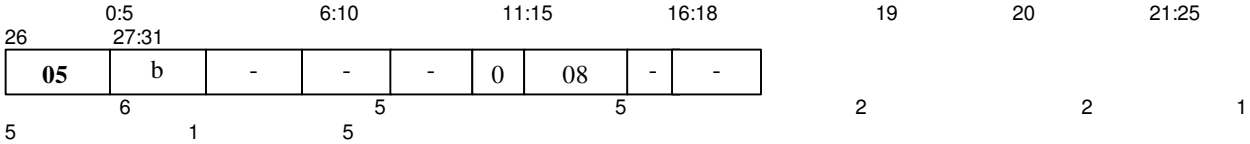
Operation :
```
entry <-- GR[b][26:31];
set       <-- GR[b][24:25];
field     <-- GR[b][23];

if (field == 0)
        GR[t][0:31] <-- Btable[set, entry][0:31];
else
        GR[t][30:31] <-- Btable[set, entry][32:33];
```

Exception : Privilege instruction trap.

Restriction : This instruction can only be executed by code running at the most privilege level.

Move to Itag

Format: MTITAG b



Purpose : To copy a value into Itag from a general register.

Description : GR[b][0:20] is copied into a specified entry of BTB_tag.

Operation : entry <-- GR[b][24:31];

Itag[entry][0:20] <-- GR[b][0:20];

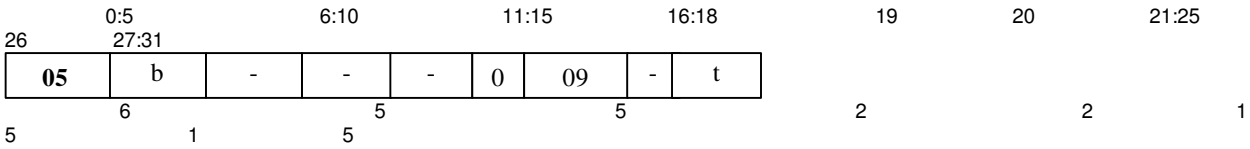
Exception : Privilege instruction trap.

Restriction : This instruction can only be executed by code running at the most privilege level.

Note : Itag[entry][0:19] : tag field
Itag[entry][20] : valid bit

Move from Itag

Format: MFITAG b,t



Purpose : To copy a value into a general register from Itag.

Description : A specified entry of Itag is copied into GR[t].

Operation : entry <-- GR[b][24:31];

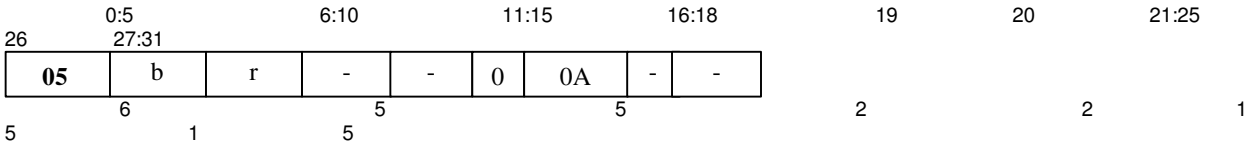
GR[t][0:20] <-- Itag[entry][0:20];

Exception : Privilege instruction trap.

Restriction : This instruction can only be executed by code running at the most privilege level.

Move to Icache

Format: MTICAH r,b



Purpose : To copy a value into Icache from a general register.

Description : GR[r] is copied into a specified entry of Icache.

Operation : entry <-- GR[b][20:27];
word <-- GR[b][28:29];

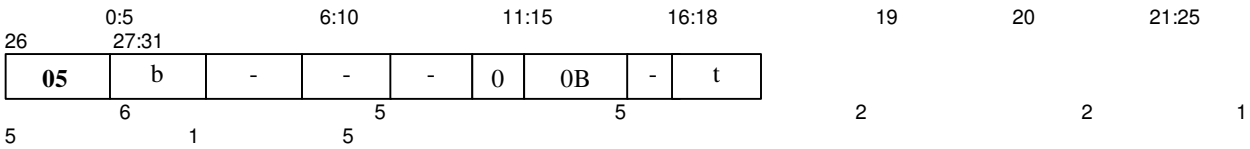
Btable[entry, word][0:31] <-- GR[r][0:31];

Exception : Privilege instruction trap.

Restriction : This instruction can only be executed by code running at the most privilege level.

Move from Icache

Format: MFICAH b,t



Purpose : To copy a value into a general register from Icache.

Description : A word of specified Icache_entry is copied into GR[t].

Operation : entry <-- GR[b][20:27];
word <-- GR[b][28:29];

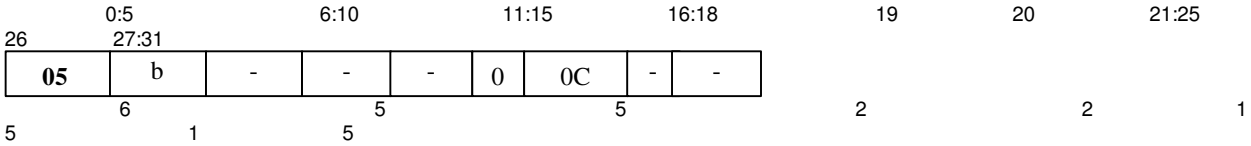
GR[t][0:31] <-- Icache[entry, word][0:31];

Exception : Privilege instruction trap.

Restriction : This instruction can only be executed by code running at the most privilege level.

Move to Dtag

Format: MTDTAG b



Purpose : To copy a value into Dtag from a general register.

Description : GR[b][0:22] is copied into a specified entry of Dtag.

Operation : entry <-- GR[b][22:27];
set <-- GR[b][30:31];

Dtag[set, entry][0:22] <-- GR[b][0:22];

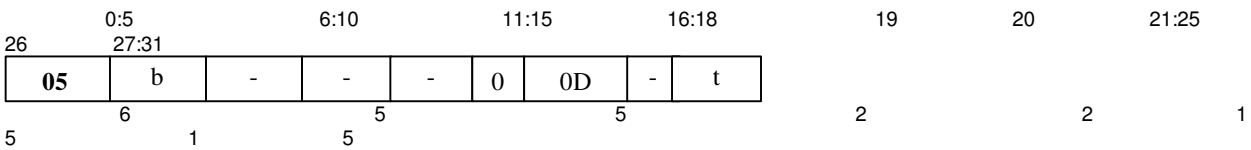
Exception : Privilege instruction trap.

Restriction : This instruction can only be executed by code running at the most privilege level.

Note : Dtag[set, entry][0:21] : tag field
Dtag[set, entry][22] : valid bit

Move from Dtag

Format: MFDTAG b,t



Purpose : To copy a value into a general register from Dtag.

Description : A specified entry of Dtag is copied into GR[t].

Operation : entry <-- GR[b][22:27];
set <-- GR[b][30:31];

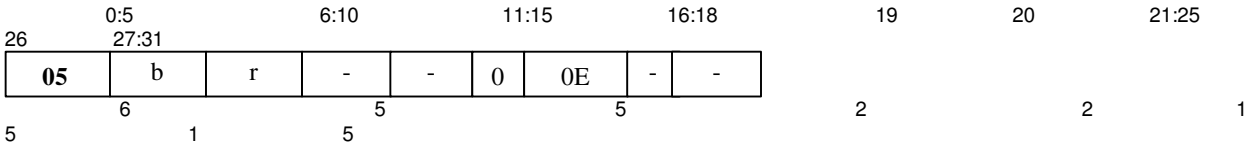
GR[t][0:22] <-- Dtag[set, entry][0:22];

Exception : Privilege instruction trap.

Restriction : This instruction can only be executed by code running at the most privilege level.

Move to Dcache

Format: MTDCAH r,b



Purpose : To copy a value into Dcache from a general register.

Description : GR[r] is copied into a specified entry of Dcache.

Operation : entry <-- GR[b][22:27];
word <-- GR[b][28:29];
set <-- GR[b][30:31];

Dcache[set, entry, word][0:33] <-- {GR[r][0:31], GR[b][20:21]};

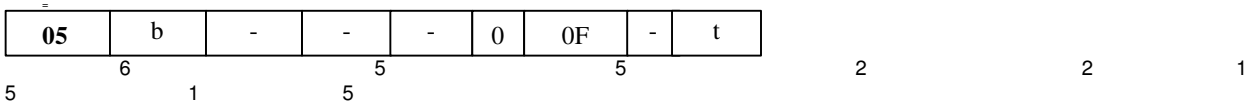
Exception : Privilege instruction trap.

Restriction : This instruction can only be executed by code running at the most privilege level.

Note : Dcache[set, entry, word][0:31] : data word field
Dcache[set, entry][32] : dirty bit
Dcache[set, entry][33] : nru bit

Move from Dcache

Format: MFDCAH b,t



Purpose : To copy a value into a general register from Dcache.

Description : A word of specified Dcache_entry is copied into GR[t].

Operation :
```
entry <-- GR[b][22:27];
word <-- GR[b][28:29];
set          <-- GR[b][30:31];
field    <-- GR[b][21];

if (field == 0)
        GR[t][0:31]      <-- Dcache[set, entry, word][0:31];
else
        GR[t][30:31] <-- Dcache[set, entry][32:33];
```

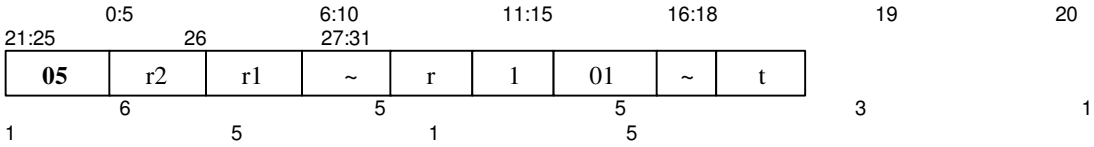
Exception : Privilege instruction trap.

Restriction : This instruction can only be executed by code running at the most privilege level.

APPENDIX C : MULTIPLIER EXTENDED INSTRUCTION SET

Halfword Multiply HMUL

Format: HMUL, cmplt r1,r2,t



Purpose: To multiply corresponding 16-bit sign halfword of two general registers.

Description: The corresponding 16-bit sign halfwords of GR[r1] and GR[r2] are arithme-tically multiplied. The multiply result is placed in 32-bit LO accumulate register and GR[t] register. The bit in AIR[25] which indicates operating in integer or fraction mode determines the high-order or low-order 16 bits of GR[r1], GR[r2] will be as the two operands.

The completer, *cmplt*, determines multiplication in rounding or unrounding mode is performed, the completer specified by "r" indicates operating in rounding mode, the multiply result can be truncated the lower 16 bits when the least 16th bit is zero. IF the the least 16th bit is one, add one with the high-order 48 bits and truncate the low-order 16 bits.

Operation:

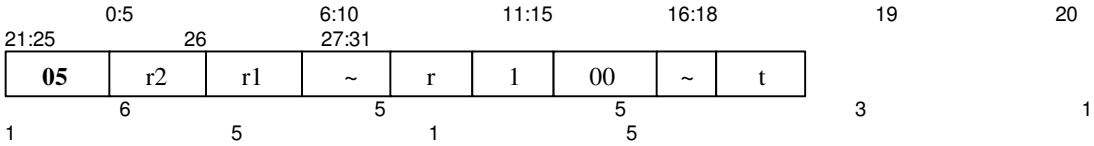
```
Integer mode operation (AIR[25] = 0) :
switch (cmplt) {
        case r : (r=1, rounding mode){
                GR[t]{0:31}  ←  (sign_ext(GR[r1]{16:31})  *
sign_ext(GR[r2]{16:31})+16h8000);
                LO{0:31}  ←  (sign_ext(GR[r1]{16:31})  *
sign_ext(GR[r2]{16:31})+16h8000);
                break;
                }
        default : (r=0,  unrounding mode){
                GR[t]{0:31} ← (sign_ext(GR[r1]{16:31}) * sign_ext(GR[r2]{16:31}));
                LO{0:31}  ← (sign_ext(GR[r1]{16:31}) * sign_ext(GR[r2]{16:31}));
                break;
                }
}

Fraction mode operation (AIR[25] = 1):
switch(cmplt){
        case r : (r=1, rounding mode){
                GR[t]{0:31} ← (lshift(sign_ext(GR[r1]{0:15}) * sign_ext(GR[r2]{0:15}), 1)
                +16h8000);
```

```
                                LO{0:31}        ← (lshift(sign_ext(GR[r1]{0:15}) * sign_ext(GR[r2]{0:15}),
1)
                                +16h8000);
                                break;
                                }
        default : (r=0, unrounding mode){
                                GR[t]{0:31} ← lshift(sign_ext(GR[r1]{0:15}) * sign_ext(GR[r2]{0:15}), 1);
                                LO{0:31}        ← lshift(sign_ext(GR[r1]{0:15}) * sign_ext(GR[r2]{0:15}),
1);
                                break;
                                }
}
```

Exception : None

Halfword Multiply Unsign HMULU

Format: HMULU, cmplt r1,r2,t



Purpose: To multiply corresponding 16-bit unsign halfword of two general registers.

Description: The corresponding 16-bit unsign halfwords of GR[r1] and GR[r2] are arithme-tically multiplied. The multiply result is placed in 32-bit LO accumulate register and GR[t] register. The bit in AIR[25] which indicates operating in integer or fraction mode determines the high-order or low-order 16 bits of GR[r1], GR[r2] will be as the two operands.

The completer, *cmplt*, determines multiplication in rounding or unrounding mode is performed, the completer specified by "r" indicates operating in rounding mode, the multiply result can be truncated the lower 16 bits when the least 16th bit is zero. IF the the least 16th bit is one, add one with the high-order 48 bits and truncate the low-order 16 bits.

Operation:

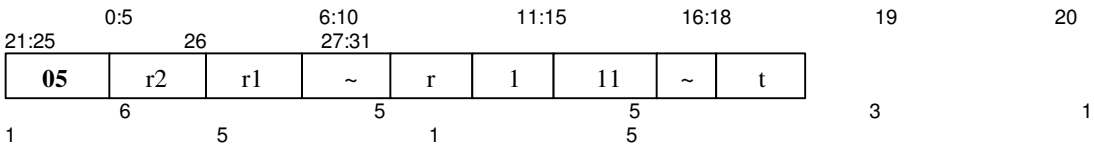
```
Integer mode operation (AIR[25] = 0) :
switch (cmplt) {
        case r : (r=1, rounding mode){
                                GR[t]{0:31}     ←       (zero_ext(GR[r1]{16:31})        *
zero_ext(GR[r2]{16:31})+16h8000);
                                LO{0:31}                ←       (zero_ext(GR[r1]{16:31})        *
zero_ext(GR[r2]{16:31})+16h8000);
                                break;
                }
```

```
        default : (r=0, unrounding mode){
                                GR[t]{0:31} ← (zero_ext(GR[r1]{16:31}) * zero_ext(GR[r2]{16:31}));
                                LO{0:31}        ← (zero_ext(GR[r1]{16:31}) * zero_ext(GR[r2]{16:31}));
                                break;
                                }
}

Fraction mode operation (AIR[25] = 1) :
switch (cmplt) {
            case r : (r=1, rounding mode){
                                GR[t]{0:31} ← (lshift(zero_ext(GR[r1]{0:15}) * zero_ext(GR[r2]{0:15}), 1)
                                +16h8000);
                                LO{0:31}        ← (lshift(zero_ext(GR[r1]{0:15}) * zero_ext(GR[r2]{0:15}),
1)
                                +16h8000);
                                break;
                                }
        default : (r=0, unrounding mode){
                                GR[t]{0:31} ← lshift(zero_ext(GR[r1]{0:15}) * zero_ext(GR[r2]{0:15}), 1);
                                LO{0:31}        ← lshift(zero_ext(GR[r1]{0:15}) * zero_ext(GR[r2]{0:15}),
1);
                                break;
                                }
}
```

Exception : None

Multiply MUL

Format: MUL, cmplt r1,r2,t



Purpose: To multiply corresponding 32-bit sign words of two general registers.

Description: The corresponding 32-bit sign words of GR[r1] and GR[r2] are arithmetically multiplied. The multiply results are placed in 64-bit {HI, LO} accumulate register and word result is placed in GR[t]. The bit in AIR[25] indicates operating in integer or fraction mode.

The completer, *cmplt*, specified by "r" indicates operating in rounding mode, the multiply result can be truncated the lower 32 bits when the least 32th bit is zero. IF the the least 32th bit is one, add one the high-order 32 bits and truncate the low-order 32 bits.

Operation:

```
Integer mode operation (AIR[25] = 0) :
switch (cmplt) {
        case r : (r=1, rounding mode){
                GR[t]{0:31} ← ((sign_ext(GR[r1]), 64) * (sign_ext(GR[r2]),
64)+32h80000000){0:31};
                {HI, LO} ← ((sign_ext(GR[r1]), 64) * (sign_ext(GR[r2],
64)+32h80000000);
                break;
                }
        default : (r=0, unrounding mode){
                GR[t]{0:31} ← ((sign_ext(GR[r1]), 64) * (sign_ext(GR[r2]), 64)){32:63};
                {HI, LO} ← ((sign_ext(GR[r1]), 64) * (sign_ext(GR[r2]), 64));
                break;
                }
}

Fraction mode operation (AIR[25] = 1) :
switch(cmplt) {
        case r : (r=1, rounding mode){
                GR[t]{0:31} ← (lshift((sign_ext(GR[r1]), 64) * (sign_ext(GR[r2]), 64), 1)
                +32h80000000){0:31};
                {HI, LO} ← (lshift((sign_ext(GR[r1]), 64) * (sign_ext(GR[r2]), 64),
1)
                +32h80000000);
                break;
                }
        default : (r=0, unrounding mode){
                GR[t]{0:31} ← lshift((sign_ext(GR[r1]), 64) * (sign_ext(GR[r2]), 64),
1){0:31};
                {HI, LO} ← lshift((sign_ext(GR[r1]), 64) * (sign_ext(GR[r2]), 64),
1);
                break;
                }
}
```

Exception : None

Multiply Unsign MULU

Format: MULU, cmplt r1,r2,t

0:5	6:10	11:15	16:18	19	20	21:25	26	27:31
05	r2	r1	~	r	1	10	~	t

6 5 5 3 1
1 5 1 5

Purpose: To multiply corresponding 32-bit sign words of two general registers.

Description: The corresponding 32-bit sign words of GR[r1] and GR[r2] are arithmetically multiplied. The multiply results are placed in 64-bit {HI, LO} accumulate register and word result is placed in GR t. The bit in AIR[25] indicates operating in integer or fraction mode.

The completer, *cmplt*, specified by "r" indicates operating in rounding mode, the multiply result can be truncated the lower 32 bits when the least 32th bit is zero. IF the the least 32th bit is one, add one the high-order 32 bits and truncate the low-order 32 bits.

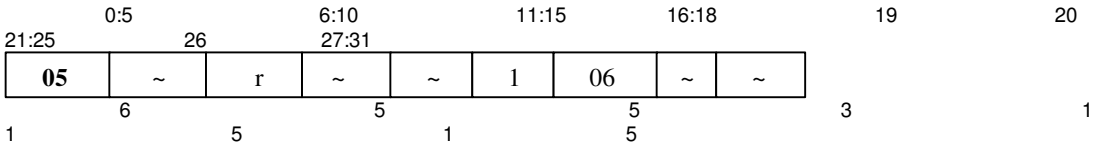
Operation:

```
Integer mode operation (AIR[25] = 0) :
switch (cmplt) {
        case r : (r=1, rounding mode){
                GR[t]{0:31} ← ((zero_ext(GR[r1]), 64) * (zero_ext(GR[r2]),
64))+32h80000000){0:31};
                {HI, LO} ← ((zero_ext(GR[r1]), 64) * (zero_ext(GR[r2]),
64)+32h80000000;
                break;
                }
        default : (r=0, unrounding mode){
                GR[t]{0:31} ← ((zero_ext(GR[r1]), 64) * (zero_ext(GR[r2]), 64)){32:63};
                {HI, LO} ← ((zero_ext(GR[r1]), 64) * (zero_ext(GR[r2]), 64));
                break;
                }
}

Fraction mode operation (AIR[25] = 1) :
switch (cmplt) {
        case r : (r=1, rounding mode){
                GR[t]{0:31} ← (lshift((zero_ext(GR[r1]), 64) * (zero_ext(GR[r2]), 64), 1)
                +32h80000000){0:31};
                {HI, LO} ← (lshift((zero_ext(GR[r1]), 64) * (zero_ext(GR[r2]), 64),
1)
                +32h80000000;
                break;
                }
        default : (r=0, unrounding mode){
                GR[t]{0:31} ← lshift((zero_ext(GR[r1]), 64) * (zero_ext(GR[r2]), 64),
1){0:31};
                {HI, LO} ← lshift((zero_ext(GR[r1]), 64) * (zero_ext(GR[r2]), 64),
1);
                break;
                }
}
```

Exception : None

Move To HI MTHI

Format: MTHI, r



Purpose: To move a general register into high-order 32-bit accumulator register HI .

Description: Load the contents of general-purpose registor GR[r] into 32 bits high-order accumulator register HI and automatic sign_extended into 4 hidden bits.

Operation :

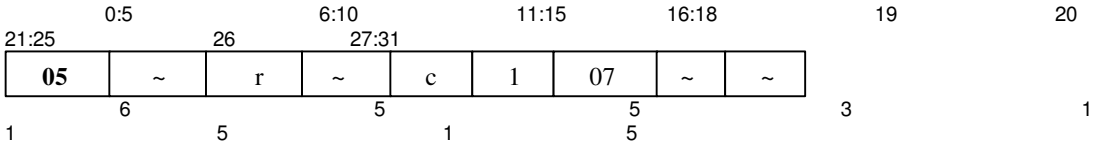
HI ← GR[r];
HIDDEN[0:3] ← {4{GR[r]{0}}};

*** HIDDEN : Hidden bits are implemented to avoid saturation in word operation. (reference SAT inatruction).**

Exception : None

Move To LO MTLO

Format: MTLO, cmplt, r



Purpose: To move a general register into low-order 32-bit accumulator register LO .

Description: Load the contents of general-purpose registor GR[r] into 32 bits low-order accumulator register. The *cmplt,* c determines whether automatic sign_extended to the 32 bits high-order accumulator HI;

Operation :

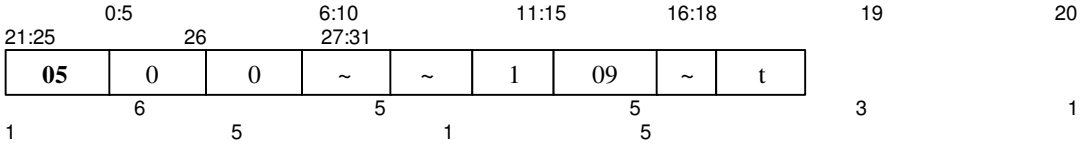
```
switch (cmplt){
        case e : (c = 1) {
```

```
                                    LO                          ← GR[r];
                                    HI                            ← {32{GR[r]{0}}};
                                    HIDDEN[0:3]    ← {4{GR[r]{0}}};
                                  }
                default : (c = 0) {
                                    LO ← GR[r];
                                  }
}
```

Exception : None

Move From HI — MFHI

Format: MFHI, t



Purpose: To move the high-order 32-bits of the accumulator HI into a general register GR[t].

Description: The high-order 32 bits accumulator register HI is stored into a general register GR[t].

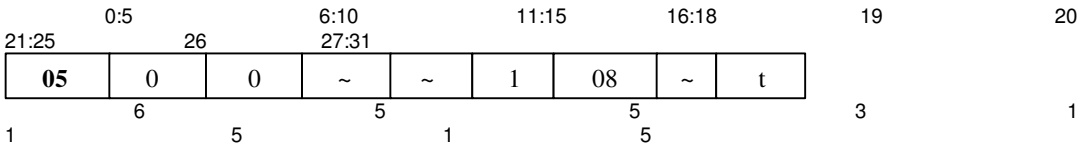
Operation :

```
GR[t] ← HI[0:31];
```

Exception : None

Move From LO — MFLO

Format: MFLO, t



Purpose: To move the low-order 32-bits accumulator register LO into a general register GR[t].

Description: The low-order 32 bits accumulator register LO is stored into a general register GR[t].

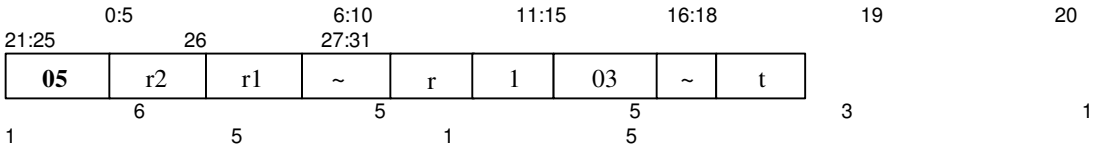
Operation :

GR[t] ← LO[0:31];

Exception : None

Halfword Multiply And Accumulate HMAC

Format: HMAC, cmplt r1,r2,t



Purpose: To multiply two signed 16-bit halfword of GR[r1] register and GR[r2] register, then accumulate {HI, LO} register with the multiplied result.

Description: The corresponding 16-bit halfwords of GR[r1] and GR[r2] are interpretted as signed 16-bit operands, and are arithmetically multiplied and add the product to the present contents of the {HI, LO} register, the 64-bit result is placed in {HI, LO} register and GR[t], the bit in AIR[25] which indicates operating in integer or fraction mode determines the high-order halfword or low-order halfword of GR[r1], GR[r2] will be as the two operands .

The completer "r" indicates operating in rounding mode, the multiply result can be truncated the lower 16 bits when the least 16th bit is zero. IF the the least 16th bit is one, add one the high-order 16 bits and truncate the low-order 16 bits.

Operation:

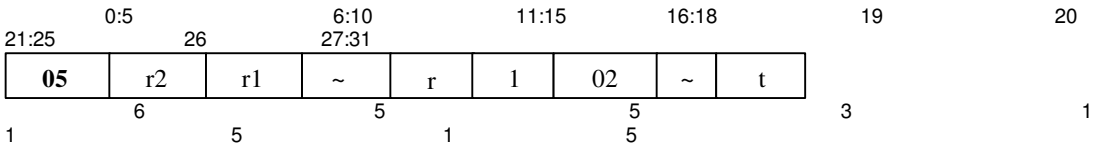
```
Integer mode operation (AIR[25] = 0) :
switch (cmplt) {
        case r: (r=1, rounding mode){
                GR[t]{0:31} ← ({HI, LO} + ((sign_ext(GR[r1]{16:31}) *
sign_ext(GR[r2]{16:31})
                                + 16h8000)){32:63};
                {HI, LO} ← ({HI, LO} + ((sign_ext(GR[r1]{16:31}) *
sign_ext(GR[r2]{16:31})
                                +16h8000)){0:63};
                break;
                }
      default : (r=0, unrounding mode){
```

```
                                GR[t]{0:31}  ←  {HI,  LO}  +  sign_ext(GR[r1]{16:31})  *
sign_ext(GR[r2]{16:31}){32:63};
                              {HI,  LO}  ←  {HI,  LO}  +  sign_ext(GR[r1]{16:31})  *
sign_ext(GR[r2]{16:31});
                                break;
                              }
}

Fraction mode opeartion (AIR[25] = 1) :
switch(cmplt){
        case r: (r=1, rounding mode){
                              GR[t]{0:31}  ←  ({HI,  LO}  +  (lshift(sign_ext(GR[r1]{0:15})  *
sign_ext(GR[r2]{0:15}),
                                                          1) + 16h8000)){32:63};
                              {HI,  LO}  ←  ({HI,  LO}  +  (lshift(sign_ext(GR[r1]{0:15})  *
sign_ext(GR[r2]{0:15}),
                                                          1) + 16h8000)){0:63};
                              break;
                              }
    default : (r=0, unrounding mode){
                              GR[t]{0:31}  ←  ({HI,  LO}  +  (lshift(sign_ext(GR[r1]{0:15})  *
sign_ext(GR[r2]{0:15}),
                                                          1){32:63};
                            {HI,  LO}  ←  ({HI,  LO}  +  (lshift(sign_ext(GR[r1]{0:15})  *
sign_ext(GR[r2]{0:15}),
                                                          1){0:63};
                              break;
                                }
}
```

Exception : None

Halfword Multiply And Accumulate Unsign HMACU

Format: HMACU, cmplt r1,r2,t



Purpose: To multiply two unsigned 16-bit halfword of GR[r1] and GR[r2] register, then accumulate {HI, LO} register with the multiplied result.

Description: The corresponding 16-bit halfwords of GR[r1] and GR[r2] are interpretted as unsigned 16-bit operands, and are arithmetically multiplied and add the product to the present

contents of the {HI, LO} register, the 64-bit result is placed in {HI, LO} register and GR[t], the bit in AIR[25] which indicates operating in integer or fraction mode determines the high-order halfword or low-order halfword of GR[r1], GR[r2] will be as the two operands .

The completer "r" indicates operating in rounding mode, the multiply result can be truncated the lower 16 bits when the least 16th bit is zero. IF the the least 16th bit is one, add one the high-order 16 bits and truncate the low-order 16 bits.

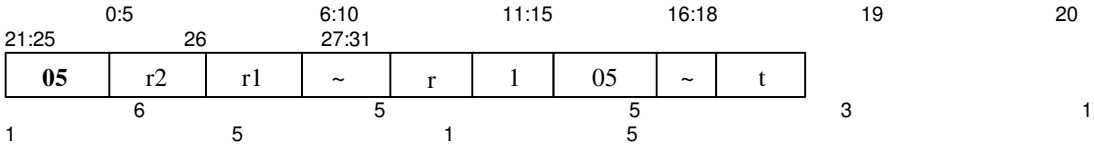
Operation:

```
Integer mode operation (AIR[25] = 0) :
switch (cmplt) {
        case r : (r=1, rounding mode){
                GR[t]{0:31} ← ({HI, LO} + ((zero_ext(GR[r1]{16:31}) *
zero_ext(GR[r2]{16:31})
                                + 16h8000)){32:63};
                {HI, LO} ← ({HI, LO} + (zero_ext(GR[r1]{16:31}) *
zero_ext(GR[r2]{16:31})
                                +16h8000)){0:63};
                break;
                }
    default : (r=0, unrounding mode){
                GR[t]{0:31} ← {HI, LO} + zero_ext(GR[r1]{16:31}) *
zero_ext(GR[r2]{16:31}){32:63};
                {HI, LO} ← {HI, LO} + zero_ext(GR[r1]{16:31}) *
zero_ext(GR[r2]{16:31});
                break;
                }
}

Fraction mode operation (AIR[25] = 1) :
switch(cmplt) {
    case r : (r=1, rounding mode){
                GR[t]{0:31} ← ({HI, LO} + lshift(zero_ext(GR[r1]{0:15}) * zero_ext(GR[r2]{0:15}),
                                1) + 16h8000){32:63};
                {HI, LO} ← ({HI, LO} + lshift(zero_ext(GR[r1]{0:15}) *
zero_ext(GR[r2]{0:15}),
                                1) + 16h8000){0:63};
                break;
                }
    default : (r=0, unrounding mode){
                GR[t]{0:31} ← ({HI, LO} + lshift(zero_ext(GR[r1]{0:15}) *
zero_ext(GR[r2]{0:15}),
                                1)){32:63};
                {HI, LO} ← ({HI, LO} + lshift(zero_ext(GR[r1]{0:15}) *
zero_ext(GR[r2]{0:15}),
                                1)){0:63};
                break;
                }
}
```

Exception : None

Halfword Multiply And Subtract HMSB

Format: HMSB, cmplt r1,r2,t



Purpose: To multiply two sign 16-bit halfword of GR[r1] and GR[r2], then subtract the multiplied result from {HI, LO} accumulate register.

Description: The corresponding 16-bit halfwords of GR[r1] and GR[r2] are interpretted as signed operands, and are arithmetically multiplied and subtract the product from the present contents of the {HI, LO} register, the 64-bit result is placed in {HI, LO} register, and word result is placed in GR[t]. The bit in AIR[25] which indicates operates in integer or fraction mode determines the high-order halfword or low-order halfword of GR[r1], GR[r2] will be as the two operands .

The completer "r" indicates operating in rounding mode, the multiply result can be truncated the lower 16 bits when the least 16th bit is zero. IF the the least 16th bit is one, add one the high-order 16 bits and truncate the low-order 16 bits.

Operation:

```
Integer mode operation (AIR[25] = 0) :
switch (cmplt) {
        case r: (r=1, rounding mode){
                GR[t]{0:31} ← ({HI, LO} - ((sign_ext(GR[r1]{16:31}) *
sign_ext(GR[r2]{16:31})
                                + 16h8000)){32:63};
                {HI, LO} ← ({HI, LO} - ((sign_ext(GR[r1]{16:31}) *
sign_ext(GR[r2]{16:31})
                                +16h8000)){0:63};
                break;
                }
      default : (r=0, unrounding mode){
                GR[t]{0:31} ← {HI, LO} - sign_ext(GR[r1]{16:31}) *
sign_ext(GR[r2]{16:31}){32:63};
                {HI, LO} ← {HI, LO} - sign_ext(GR[r1]{16:31}) *
sign_ext(GR[r2]{16:31});
                break;
                }
}
```

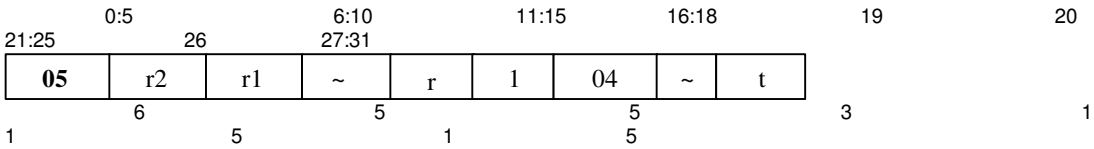
Fraction mode opeartion (AIR[25] = 1) :

```
switch(cmplt){
        case r: (r=1, rounding mode){
                        GR[t]{0:31} ← ({HI, LO} - (lshift(sign_ext(GR[r1]{0:15}) * sign_ext(GR[r2]{0:15}),
                                                                        1) + 16h8000)){32:63};
                        {HI,    LO}             ←   ({HI,   LO}   -   (lshift(sign_ext(GR[r1]{0:15})   *
sign_ext(GR[r2]{0:15}),
                                                                        1) + 16h8000)){0:63};
                        break;
                        }
    default : (r=0, unrounding mode){
                        GR[t]{0:31} ← ({HI, LO} - (lshift(sign_ext(GR[r1]{0:15}) * sign_ext(GR[r2]{0:15}),
                                                                        1){32:63};
                       {HI,    LO}             ←   ({HI,   LO}   -   (lshift(sign_ext(GR[r1]{0:15})   *
sign_ext(GR[r2]{0:15}),
                                                                        1){0:63};
                        break;
                        }
}
```

Exception : None

Halfword Multiply And Subtract Unsign HMSBU

Format: HMSBU, cmplt r1,r2,t



Purpose: To multiply two unsigned 16-bit halfword of GR[r1] and GR[r2] , then subtract the multiplied result from {HI, LO} accumulate register..

Description: The corresponding 16-bit halfwords of GR[r1] and GR[r2] are interpretted as unsigned 16-bit operands, and are arithmetically multiplied, then subtract the product from the present contents of the {HI, LO} register, the 64-bit result is placed in {HI, LO} register and GR[t], the bit in AIR[25] which indicates operating in integer or fraction mode determines the high-order halfword or low-order halfword of GR[r1], GR[r2] will be as the two operands .

The completer "r" indicates operating in rounding mode, the multiply result can be truncated the lower 16 bits when the least 16th bit is zero. IF the the least 16th bit is one, add one the high-order 16 bits and truncate the low-order 16 bits.

Operation:

```
Integer mode operation (AIR[25] = 0) :
switch (cmplt) {
```

```
        case r : (r=1, rounding mode){
                GR[t]{0:31}  ←  ({HI,  LO}  -  ((zero_ext(GR[r1]{16:31})  *
zero_ext(GR[r2]{16:31})
                                                + 16h8000)){32:63};
                {HI,  LO}  ←  ({HI,  LO}  -  (zero_ext(GR[r1]{16:31})  *
zero_ext(GR[r2]{16:31})
                                                +16h8000)){0:63};
                break;
                }
        default : (r=0, unrounding mode){
                GR[t]{0:31}  ←  {HI,  LO}  -  zero_ext(GR[r1]{16:31})  *
zero_ext(GR[r2]{16:31}){32:63};
                {HI,  LO}  ←  {HI,  LO}  -  zero_ext(GR[r1]{16:31})  *
zero_ext(GR[r2]{16:31});
                break;
                }
}

Fraction mode operation (AIR[25] = 1) :
switch(cmplt) {
        case r : (r=1, rounding mode){
                GR[t]{0:31} ← ({HI, LO} - lshift(zero_ext(GR[r1]{0:15}) * zero_ext(GR[r2]{0:15}),
                                                1) + 16h8000){32:63};
                {HI,  LO}  ←  ({HI,  LO}  -  lshift(zero_ext(GR[r1]{0:15})  *
zero_ext(GR[r2]{0:15}),
                                                1) + 16h8000){0:63};
                break;
                }
        default : (r=0, unrounding mode){
                GR[t]{0:31}  ←  ({HI,  LO}  -  lshift(zero_ext(GR[r1]{0:15})  *
zero_ext(GR[r2]{0:15}),
                                                1)){32:63};
                {HI,  LO}  ←  ({HI,  LO}  -  lshift(zero_ext(GR[r1]{0:15})  *
zero_ext(GR[r2]{0:15}),
                                                1)){0:63};
                break;
                }
}
```

Exception : None

Multiply And Accumulate — MAC

Format: MAC, cmplt r1,r2,t

0:5	6:10	11:1	16:18	19	20	21:25	26	27:31
05	r2	r1	~	r	1	13	~	t

6 5 5 3 1
1 5 1 5

Purpose: To multiply two sign 32-bit word of GR[r1] and GR[r2], then add accumulate register {HI, LO} with the multiplied result.

Description: The corresponding 32-bit word of GR[r1] and GR[r2] are interpretted as signed 32-bit operands, and are arithmetically multiplied and add the product with the present contents of the {HI, LO} register, the 64-bit result is placed in {HI, LO} register. The bit in AIR[25] indicates operating in integer or fraction mode.

The completer "r" indicates operating in rounding mode, the multiply result can be truncated the lower 32 bits when the least 32th bit is zero. IF the the least 32th bit is one, add one the high-order 32 bits and truncate the low-order 32 bits.

Operation:

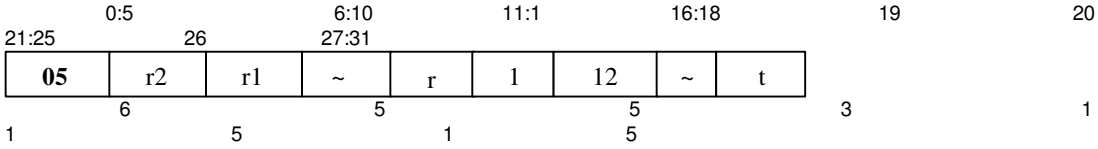
```
Integer mode operation (AIR[25] = 0) :
switch (cmplt) {
        case r : (r=1, rounding mode){
                GR[t]{0:31} ← ({HI, LO} + (sign_ext(GR[r1]{0:31}) *
sign_ext(GR[r2]{0:31})
                                + 32h80000000){0:31};
                {HI, LO} ← ({HI, LO} + (sign_ext(GR[r1]{0:31}) *
sign_ext(GR[r2]{0:31})
                                + 32h80000000){0:63};
                  break;
                }
      default : (r=0, unrounding mode){
                GR[t]{0:31} ← {HI, LO} + sign_ext(GR[r1]{0:31}) *
sign_ext(GR[r2]{0:31}){32:63};
                {HI, LO} ← {HI, LO} + sign_ext(GR[r1]{0:31}) *
sign_ext(GR[r2]{0:31});
                  break;
                }
}

Fraction mode operation (AIR[25] = 1) :
switch (cmplt) {
      case r : (r=1, rounding mode){
                GR[t]{0:31} ← ({HI, LO} + lshift(sign_ext(GR[r1]{0:31}) * sign_ext(GR[r2]{0:31}),
                                1) + 32h80000000){0:31};
                {HI, LO} ← ({HI, LO} + lshift(sign_ext(GR[r1]{0:31}) *
sign_ext(GR[r2]{0:31}),
                                1) + 32h80000000){0:63};
                  break;
                }
     default : (r=0, urounding mode){
                GR[t]{0:31} ← ({HI, LO} + lshift(sign_ext(GR[r1]{0:31}) *
sign_ext(GR[r2]{0:31}),
                                1){0:31};
```

{HI, LO} ← ({HI, LO} + lshift(sign_ext(GR[r1]{0:31}) * sign_ext(GR[r2]{0:31}),
1){0:63};
break;
}
}

Exception : None

Multiply And Accumulate Unsign MACU

Format: MACU, cmplt r1,r2,t



Purpose: To multiply two unsign 32-bit word of GR[r1] and GR[r2], then add accumulate register {HI, LO} with the multiplied result.

Description: The corresponding 32-bit word of GR[r1] and GR[r2] are interpretted as unsigned 32-bit operands, and are arithmetically multiplied and add the product with the present contents of the {HI, LO} register, the 64-bit result is placed in {HI, LO} register. The bit in AIR[25] indicates operating in integer or fraction mode.

The completer "r" indicates operating in rounding mode, the multiply result can be truncated the lower 32 bits when the least 32th bit is zero. IF the the least 32th bit is one, add one the high-order 32 bits and truncate the low-order 32 bits.

Operation:

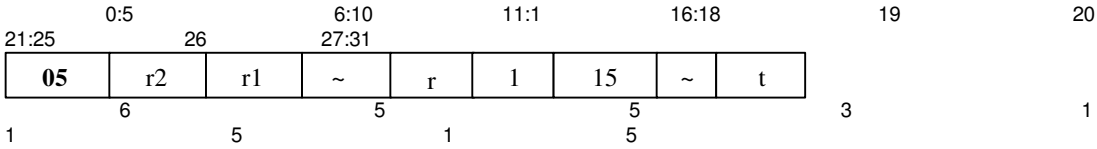
Integer mode operation (AIR[25] = 0) :
switch (cmplt) {
case r : (r=1, rounding mode){
GR[t]{0:31} ← ({HI, LO} + (zero_ext(GR[r1]{0:31}) * zero_ext(GR[r2]{0:31})
+ 32h80000000)){0:31};
{HI, LO} ← ({HI, LO} + (zero_ext(GR[r1]{0:31}) * zero_ext(GR[r2]{0:31})
+32h80000000)){0:63};
break;
}
default : (r=0, unrounding mode){
GR[t]{0:31} ← {HI, LO} + zero_ext(GR[r1]{0:31}) * zero_ext(GR[r2]{0:31}){32:63};
{HI, LO} ← {HI, LO} + zero_ext(GR[r1]{0:31}) * zero_ext(GR[r2]{0:31});
break;

```
                                }
}

Fraction mode operation :
switch(cmplt) {
        case r : (r=1, rounding mode){
                                GR[t]{0:31}  ←  ({HI,  LO}  +  lshift(zero_ext(GR[r1]{0:31})  *
zero_ext(GR[r2]{0:31}),
                                                                1) + 32h80000000){0:31};
                                {HI,  LO}  ←  ({HI,  LO}  +  lshift(zero_ext(GR[r1]{0:31})  *
zero_ext(GR[r2]{0:31}),
                                                                1) + 32h80000000){0:63};
                                  break;
                                }
      default : (r=0, unrounding mode){
                                GR[t]{0:31}  ←  ({HI,  LO}  +  lshift(zero_ext(GR[r1]{0:31})  *
zero_ext(GR[r2]{0:31}),
                                                               1){0:31};
                                {HI,  LO}  ←  ({HI,  LO}  +  lshift(zero_ext(GR[r1]{0:31})  *
zero_ext(GR[r2]{0:31}),
                                                               1){0:63};
                                  break;
                                }
}
```

Exception : None

Multiply And Subtract MSB

Format: MSB, cmplt r1,r2,t



Purpose: To multiply two sign 32-bit word of GR[r1] and GR[r2], then subtract the product from accumulate register {HI, LO}.

Description: The corresponding 32-bit word of GR[r1] and GR[r2] are interpretted as signed 32-bit operands, and are arithmetically multiplied and subtract the product from the present contents of the {HI, LO} register, the 64-bit result is placed in {HI, LO} register. The bit in AIR[25] indicates operating in integer or fraction mode.

The completer "r" indicates operating in rounding mode, the multiply result can be truncated the lower 32 bits when the least 32th bit is zero. IF the the least 32th bit is one, add one the high-order 32 bits and truncate the low-order 32 bits.

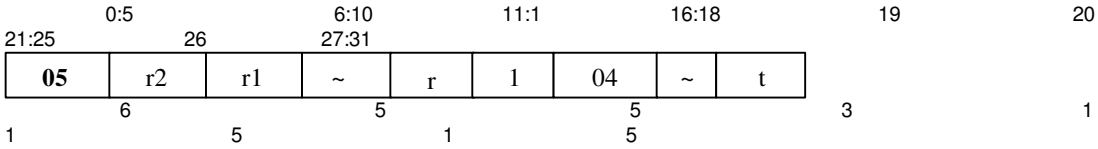
Operation:

```
Integer mode operation (AIR[25] = 0) :
switch (cmplt) {
        case r : (r=1, rounding mode){
                GR[t]{0:31} ← ({HI, LO} - (sign_ext(GR[r1]{0:31}) * sign_ext(GR[r2]{0:31})
                                                + 32h80000000){0:31};
                {HI,  LO}          ← ({HI,  LO}  -  (sign_ext(GR[r1]{0:31})  *
sign_ext(GR[r2]{0:31})
                                                + 32h80000000){0:63};
                  break;
                }
    default : (r=0, unrounding mode){
                GR[t]{0:31}  ←  {HI,  LO}  -  sign_ext(GR[r1]{0:31})  *
sign_ext(GR[r2]{0:31}){32:63};
                {HI,  LO}          ← {HI,  LO}  -  sign_ext(GR[r1]{0:31})  *
sign_ext(GR[r2]{0:31});
                  break;
                }
}

Fraction mode operation (AIR[25] = 1) :
switch (cmplt) {
     case r : (r=1, rounding mode){
                GR[t]{0:31} ← ({HI, LO} - lshift(sign_ext(GR[r1]{0:31}) * sign_ext(GR[r2]{0:31}),
                                                1) + 32h80000000){0:31};
                {HI,  LO}          ← ({HI,  LO}  -  lshift(sign_ext(GR[r1]{0:31})  *
sign_ext(GR[r2]{0:31}),
                                                1) + 32h80000000){0:31};
                  break;
                }
    default : (r=0, urounding mode){
                  GR[t]{0:31}  ←  ({HI,  LO}  -  lshift(sign_ext(GR[r1]{0:31})  *
sign_ext(GR[r2]{0:31}),
                                                1){0:31};
                  {HI,  LO}          ← ({HI,  LO}  -  lshift(sign_ext(GR[r1]{0:31})  *
sign_ext(GR[r2]{0:31}),
                                                1){0:63};
                    break;
                }
}
```

Exception : None

Multiply And Subtract Unsign MSBU

Format: MSBU, cmplt r1,r2,t



Purpose: To multiply two unsign 32-bit word of GR[r1] and GR[r2], then add accumulate register {HI, LO} with the multiplied result.

Description: The corresponding 32-bit word of GR[r1] and GR[r2] are interpretted as unsigned 32-bit operands, and are arithmetically multiplied and add the product with the present contents of the {HI, LO} register, the 64-bit result is placed in {HI, LO} register. The bit in AIR[25] indicates operating in integer or fraction mode.

The completer "r" indicates operating in rounding mode, the multiply result can be truncated the lower 32 bits when the least 32th bit is zero. IF the the least 32th bit is one, add one the high-order 32 bits and truncate the low-order 32 bits.

Operation:

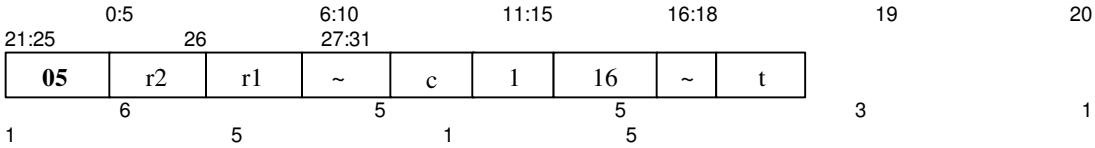
```
Integer mode operation (AIR[25] = 0) :
switch (cmplt) {
        case r : (r=1, rounding mode){
                GR[t]{0:31} ← ({HI, LO} - (zero_ext(GR[r1]{0:31}) * zero_ext(GR[r2]{0:31})
                                + 32h80000000)){0:31};
                {HI, LO} ← ({HI, LO} - (zero_ext(GR[r1]{0:31}) * zero_ext(GR[r2]{0:31})
                                +32h80000000)){0:63};
                break;
               }
    default : (r=0, unrounding mode){
                GR[t]{0:31} ← {HI, LO} - zero_ext(GR[r1]{0:31}) * zero_ext(GR[r2]{0:31}){32:63};
                {HI, LO} ← {HI, LO} - zero_ext(GR[r1]{0:31}) * zero_ext(GR[r2]{0:31});
                break;
              }
}

Fraction mode operation :
switch(cmplt) {
        case r : (r=1, rounding mode){
                GR[t]{0:31} ← ({HI, LO} - lshift(zero_ext(GR[r1]{0:31}) * zero_ext(GR[r2]{0:31}),
                                1) + 32h80000000){0:31};
                {HI, LO} ← ({HI, LO} - lshift(zero_ext(GR[r1]{0:31}) * zero_ext(GR[r2]{0:31}),
                                1) + 32h80000000){0:63};
                break;
               }
    default : (r=0, unrounding mode){
```

```
                    GR[t]{0:31}  ←  ({HI,  LO}  -  lshift(zero_ext(GR[r1]{0:31})  *
zero_ext(GR[r2]{0:31}),
                                                          1){0:31};
                    {HI,  LO}  ←  ({HI,  LO}  -  lshift(zero_ext(GR[r1]{0:31})  *
zero_ext(GR[r2]{0:31}),
                                                          1){0:63};
                    break;
                  }
}
```

Exception : None

Saturation SAT

Format: SAT, cmplt, t



Purpose: To test accumulate register {HI, LO} overflow, then saturate the accumulate register {HI, LO}

Description: To test accumulate register {HI, LO} overflow, and if the accumulate register {HI, LO} is overflow, then saturate the accumulate register, the accumulate register is not overflow, then perform a null operation.

The saturation instruction is intended to be used at the completion of a series of multiply/accumulate operations, so that temporary overflows do **NOT** occur the accumulator to saturate.

The saturation condition and result depend on operation type of halfword or word multiply/accumulate. If half word operation, test the high-order 33-bits of the accumulate register. The possible results after execution of saturation instruction is shown in Table I. If word operation, test the carry bit (hidden 4-bits) and sign bit, the possible results after execution of saturation instruction is shown in Table II.

The *cmplt* completer c specifies the saturation mode on halfword or word operation

Table I. (Halfword saturation operation mode).
*HMV HI[0]

0	0	No change	
0	1	No change	
1	0	64	0000_0000_7FFF_FFFF
1	1	64	FFFF_FFFF_8000_0000

*HMV : HMV is set when the HI[0:31] and sign-bit LO[0] are not all the same.

Table II. (Word saturation operation mode).

*WMV	*HIDDEN[0]		
0	0	No change	
0	1	No change	
1	0	64	7FFF_FFFF_FFFF_FFFF
1	1	64	8000 _0000 _0000 _0000

*WMV : WMV is set when the hidden bits and sign-bit HI[0] are not all the same.
*HIDDEN[0] : The most significiant bit of hidden bits.

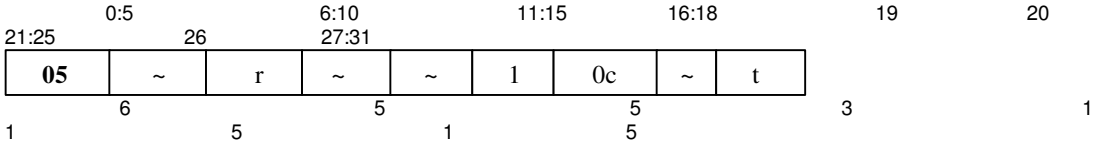
Operation:

```
switch (cmplt) {
        case h : (c = 1; word saturation){
                if (HMV) {HI, LO}← {{33{HI[0]}}, {31{~HI[0]}}};
                        GR[t] ← {HI[0], {31{~HI[0]}}};
                else
                        GR[t] ← LO;
            break;
        }
        default : (c = 0; double word saturation){
                if (WMV) {HI, LO} ← {HIDDEN[0], {63{~HIDDEN[0]}}};
                        GR[t] ← {HIDDEN[0],
{31{~HIDDEN[0]}}};
                else
                        GR[t] ← HI;
                break;
        }
}
```

Exception : None

Saturation GR[r] SATGR

Format: SATGR r, t



Purpose: To test general register GR[r] 16-bit overflow, then saturate GR[r] into G[t], GR r and Gr t may be the same register

Description: To test general GR[r] 16-bit overflow, and if the general register GR[r] is overflow, then saturate the general register GR[r], and the saturation result is put into destination register Gr[t]. If Gr[r] is not overflow, the GR[r] value is copied into GR[t].

The saturation instruction for GR[r] is intended to be used in the 16-bit halfword operation, and the result is stored in GR r register. The operation tests the high-order 32-bits of the register GR[r], and the possible results after execution of saturation is shown in TableI.

Taable I..
*HOV GR[r]{0}

0	0	No change	
0	1		No change
1	0	32h0000_7FFF	
1	1		32hFFFF_8000

*HOV : HOV is set when the high-order 17-bits of GR[r] register are not all the same.
Operation:

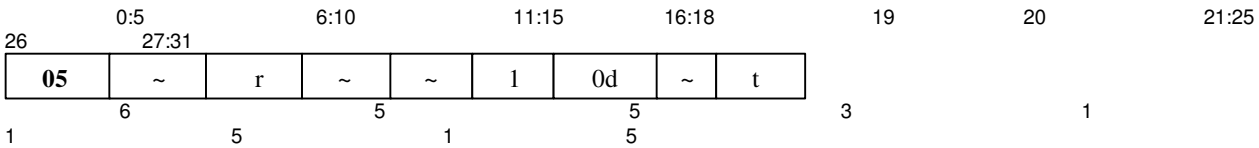
```
if(HOV) GR[t]{0:31} ← {{17{GR[r]{0}}}, {15{~GR[r]{0}}}};
break;
```

Exception : None

* Note : 16-bit halfword operation in PA-Architecture 32-bit operation. The MSB 16-bits should be sign_extented previously.

Derive Exponent EXP

Format: EXP, r, t



Purpose: To derive the effective exponent of the operand GR[r] into GR[t].

Description: The EXP operation derives the effective exponent of the input operand to prepare for normalization. Generally speaking, Normalization can be divided into two-stage. The first stage derives the effective exponent. The second stage does the actually shifting. The first uses EXP instruction which detects the exponent value and load it into GR[t]. The second stage uses the shift relative instruction to shift GR[r] by the shift amount of GR[t].

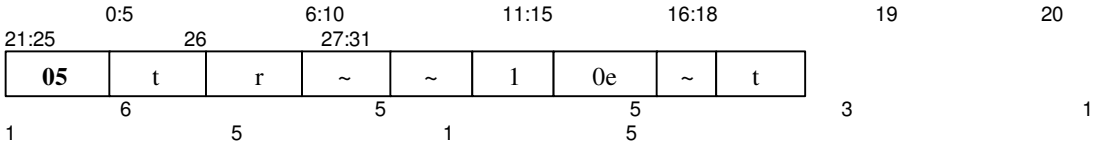
Operation:

GR[t] ← Leading sign bits of GR[r].

Exception : None

Block Exponent Adjust ADJEXP

Format: ADJEXP, r, t



Purpose: To perform on a series of numbers, derive the effective exponent of the number largest in magnitude.

Description: This function detects the effective exponent value of the number largest in magnitude in an array numbers. GR[t] remains the largest in magnitude in the effective exponent value of all previous numbers. Then compare exponent in magnitude of GR[r] with GR[t], if the effective exponent of current value (GR[r]) is smaller than the largest magnitude (GR[t]) in all previous numbers, update the GR[t] with the effective exponent value of GR[r]. Otherwise, remain the GR[t].

Operation:

```
if (EXP GR[r] < GR[t])
        GR[t] ← EXP GR[r];
```

Exception : None