MOTOROLA
SEMICONDUCTOR TECHNICAL DATA

MPA1000 Product Description

Motorola Programmable Array (MPA) products are a high density, high performance, low cost, solution for your reconfigurable logic needs. When used with our automatic high performance design tools, MPA delivers custom logic solutions in minutes rather than weeks. And the low cost keeps those solutions competitive throughout the product lifecycle.

The MPA architecture has solved the historical problems associated with fine grain devices without sacrificing re–programmability, reliability, or cost. MPA1000 devices are reprogrammable SRAM based products manufactured on a standard 0.43µ Leff CMOS process with logic capacities from 3,500 to more than 22,000 equivalent FPGA gates. MPA logic resources hold a single gate or storage element providing a highly efficient, adaptable, design implementation medium. Gate level logic resources, abundant hierarchical interconnection resources and automatic, timing driven, tools work together to quickly provide design implementations that meet timing constraints without sacrificing device utilization.

Staying focused on end product design rather than implementation tools or device architecture gets the design done faster and, unlike other programmable solutions, without programmable logic device specificity to impede future design migration efforts. The combination of automatic tools and gate level architecture is ideal for traditional schematic driven or high level language based design methodologies. In fact, logic synthesis tools were originally designed for and produce the most efficient results when targeting gate level devices.

High MPA1000 register count and controlled clock skew is ideal for designs employing pipelining techniques such as communications. The unique set of MPA1000 I/O programming options make these devices suitable for industrial and computer Interfacing circuits.

Features

- Multiple I/O from 80–200 I/O Pins
- Programmable 3V/5V I/O at Any Site
- Multiple Packaging Options
- Fine Grain Structure Is Optimized for Logic Synthesis
- Programmable Output Drive, 4/6mA @ 5.0V and 3.3V
- High Register Count, with 560–2,900 Flip–Flops
- IEEE 1149.1 JTAG Boundary Scan
- Eight Low–Skew (<1ns) Clocks

Table 2–1. MPA1000 Family Members

FPGA Gates*	Part No.	Logic Cells	Internal Flip–Flops	I/O Cell Flip–Flops	Avail I/O Pins	Packages	Availability
3500	MPA1016FN MPA1016DD	1600	400	122 160	61 80	84 PLCC 128 PQFP	NOW NOW
8000	MPA1036FN MPA1036DD MPA1036DH MPA1036HI	3600	900	122 160 240 240	61 80 120 120	84 PLCC 128 PQFP 160 PQFP 181 PGA	NOW NOW NOW NOW
14200	MPA1064DH MPA1064DK MPA1064KE MPA1064BG	6400	1600	240 320 320 320	120 160 160 160	160 PQFP 208 PQFP 224 PGA 256 PBGA	NOW NOW NOW 3Q97
22000	MPA1100DK MPA1100HV MPA1100BG	10000	2500	320 400 400	160 200 200	208 PQFP 299 PGA 256 PBGA	NOW NOW 3Q97

* Equivalent to Industry Standards, as supplied by most manufacturers.

MPA1000 Serial EPROM/EEPROM Family

Capacity	MPA Companion	Part Number	Packages	Availability	Notes
64K	MPA1016	MPA1765P MPA1765D MPA1765FN	8 DIP 8 SOIC 20 PLCC	NOW	OTP
128K	MPA1036	MPA17128P MPA17128D MPA17128FN	8 DIP 8 SOIC 20 PLCC	NOW	OTP
256K	MPA1064	MPA17C256P MPA17C256DW MPA17C256FN	8 DIP 20 SOIC 20 PLCC	3Q97	Eraseable

MPA1000 Capacity

Programmable logic gate capacity is difficult to ascertain because it is design and design tool dependent. Programmable logic capacities can only be meaningfully compared using identical designs and automatic tools. Figure 2–1 shows that under these circumstances, the MPA1036 contains from 2.1 to 1.3 XC3190 devices.

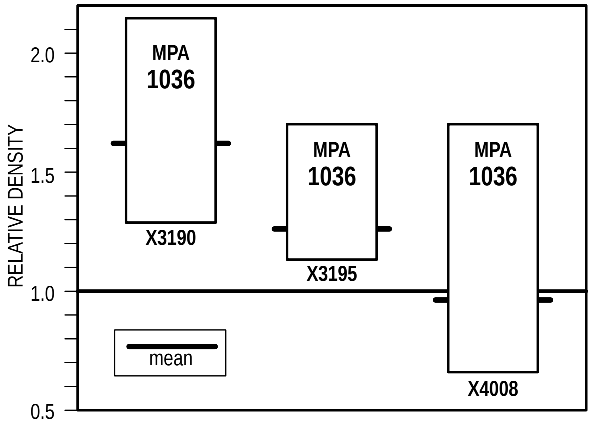


Figure 2–1. Equivalent Gate Capacity

Table 2–1 on page 2–1 shows the members of the MPA1000 family and lists the I/O, logic cell, flip flop and gate capacities for each device. To facilitate Customer device selection, Motorola rates MPA device capacity in FPGA equivalent gates. The equivalent gate counts shown were derived using identical designs and a push button implementation methodology. While this method is useful in a comparative sense, actual device capacity remains a design dependent quantity. Designs with high register gate or XOR gate to total gate ratio will pack more efficiently than the averages shown in Figure 2–1.

MPA1000 Performance

Device performance is more design and design tool dependent than device capacity. Table 2–2 shows selected cell performance figures for a typical ungraded MPA1000 device. Calculating MPA1000 DFF toggle rate from this information yields an unrealistically high expectation for device performance. Some manufacturers publish specifications for small functional blocks like counters. While more useful than toggle rates, they are based on ideal placement and routing conditions seldom achievable without manual intervention. Industry benchmarks are useful for relative comparisons of benchmark design performance, but benchmark designs don't end up in products. In addition, the design methodology used requires, manual, architecture dependent, design optimization and expert level architectural and design tool experience. Using this design methodology for real designs means a costly learning curve, severe technology migration limitations and many hours of extra design effort for each end product. If the incentive to use a programmable solution is time to market and product flexibility, this is not the ideal approach. A push button, gate level, approach increases design flexibility and improves time to market. The MPA1000 and MPA design system have been engineered to deliver a high performance gate level solution. Gate level design is widely understood, technology independent and synthesis friendly. A library of common MSI functions with optimized gate level representations are provided to reduce design implementation time.

Table 2–2. Selected MPA1000 Performance Figures

	Typical
MEDIUM BUS DELAY	1.2ns
DFF CLK TO Q	0.6ns
DFF SETUP TIME	1.5ns
TYPICAL DFF TOGGLE RATE	256MHz

(25° C, V_{CC} = 5V)

If identical designs and timing constraints are used with automatic, timing driven, design tools, a more appropriate performance comparison can be made. Figure 2–2 compares the MPA1036 vs. the XC4008 for 7 designs. The typical MPA1036 device is 48% faster than the XC4008–6 and 28% faster than the XC4008–4 for 7 identical, complex, chip level designs. In real design situations, gate level flexibility and hierarchical routing coupled with sophisticated, timing driven, design tools results in significant performance gains and reduced time to market.

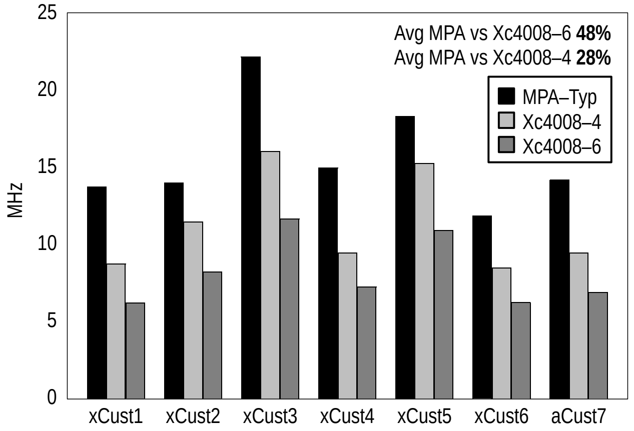


Figure 2–2. MPA1036 versus XC4008 – 7 Push Button Designs

If step and repeat style designs typical of industry benchmarks are used (Figure 2–3), MPA retains it's performance edge. While the performance gap shrinks by about 10%, absolute design performance increases dramatically compared to those shown in Figure 2–2. As critical path depth decreases, design performance increases as expected. In general these benchmarks tend to have narrowly distributed performance constraints and shallow path depths atypical of many real design implementations. In either case using benchmark information to estimate product performance for arbitrary designs is unlikely to yield reliable results. This information is intended to illustrate the range of performance enhancement possible when MPA is selected.

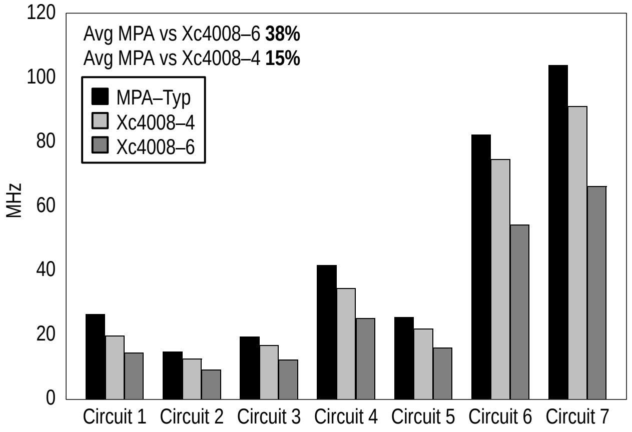


Figure 2–3. MPA1036 versus XC4008 – 7 Push Button, Step & Repeat Designs

MPA1000 Architectural Overview

MPA1000 Architecture

MPA1000 is a high density, high performance, low cost device family which maximizes application flexibility and minimizes time to market by delivering a gate level reprogrammable logic solution. Combined with automatic high performance design tools, the MPA1000 family is ideally suited to logic synthesis or gate level (gate array like) design methods.

Logic resources in the MPA1000 are fine grained – each logic cell holds a single gate or a storage element. This provides a highly efficient, adaptable, design implementation medium. Gate level logic resources, abundant hierarchical interconnection resources and automatic, timing driven, tools work together to quickly provide design implementations that meet timing constraints without sacrificing device utilization.

The MPA1000 architecture has solved the historical problems associated with fine grain architectures without sacrificing re–programmability, reliability, or cost. Previous reprogrammable fine grain architectures utilized routing architectures substantially similar to that of coarse grained products. Other fine grained architectures resorted to antifuse programming elements to address performance issues, increasing cost, while reducing reliability and abandoning reconfigurability. MPA utilizes a new routing structure which takes advantage of fine logic block granularity to achieve superior design performance.

MPA1000 devices are manufactured using a standard submicron CMOS process. SRAM cells comprise device configuration memory. MPA1000 devices can be quickly and infinitely reprogrammed.

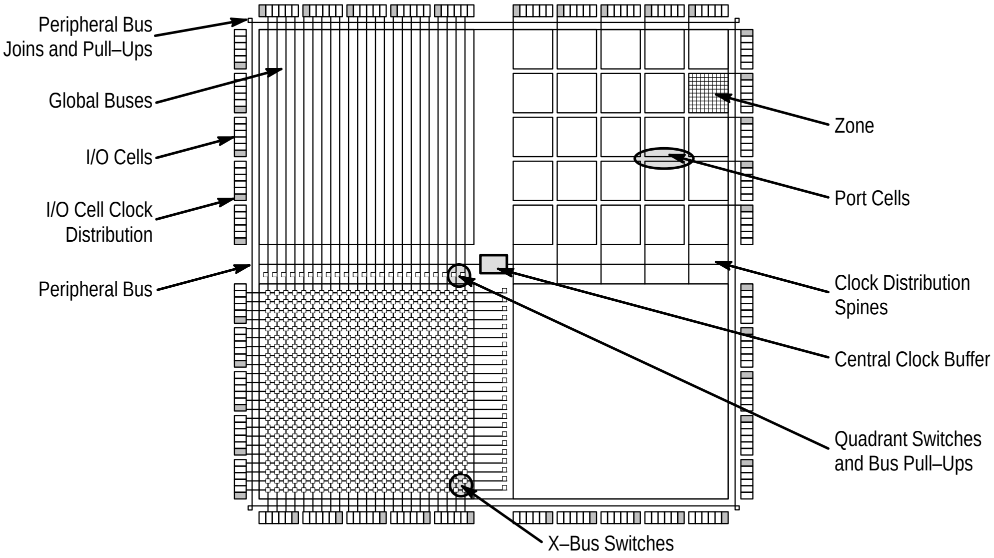


Figure 2–4. MPA Architectural Overview

Partitioned Resources

Each device is a multilevel partitioned array of cells. At the highest level of hierarchy each device is partitioned into 4 equal sized sections called quadrants. I/O cells surround the quadrants. Each quadrant is further subdivided into zones. A zone consists of a 10x10 array of core cells, 20 port cells and a clock distribution cell (Figure 2–6). Zone core cells are organized into 2x2 groups called tiles. The number of zones per quadrant defines a particular device as shown in Figure 2–5. Partitioning the device in this manner minimizes bus loading and provides an opportunity to segment device level placement and routing. This speeds design implementation time, especially if multiple processors are used. Figure 2–4 is a synopsis of the overall MPA structure.

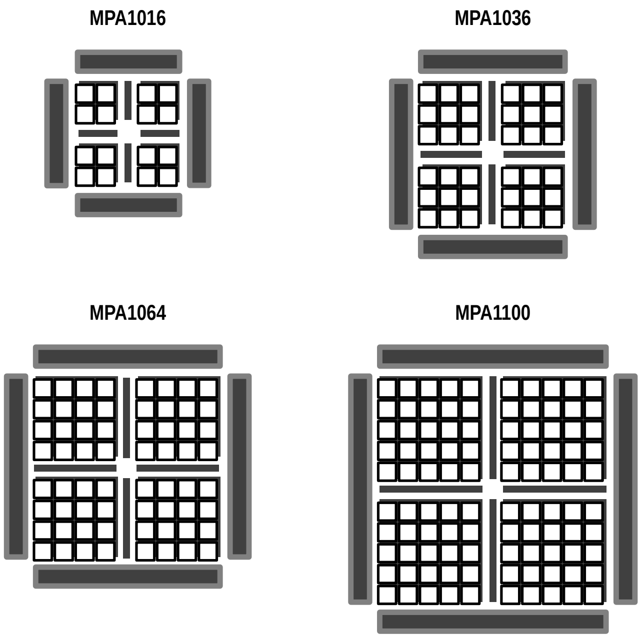


Figure 2–5. The MPA 1016, MPA1036, MPA1064 and MPA1100

Hierarchical functionality complements the robust routing resource to deliver extremely efficient design realizations. While the look up table approach of non–gate level devices can provide any function of its inputs, this flexibility is costly when simple functions are required. In contrast the simplicity, small size, and hierarchical organization of the MPA1000 delivers a more silicon efficient implementation. Logic blocks of arbitrary size and aspect ratio are automatically constructed, optimized and interconnected based on design constraints and gate level design representations. This capability complements logic synthesis technology and maximizes design migration potential. As FPGA device capacity increases, design diversity will also increase. The malleable granularity and adjustable routing resource of the MPA can accommodate this diversity with consistent silicon efficiency and performance.

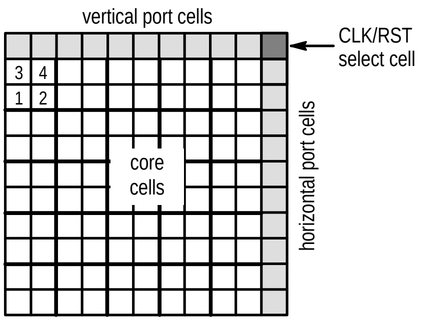


Figure 2–6. Zone Structure

Core Cells

Each core cell has 2 inputs, each input is configured to receive signals from 1 of 7 potential sources (Figure 2–7). 5 sources are from local interconnect and 2 are from zone level interconnect. Each cell output connects to 8 other cells via local interconnect and is configured to connect to up to 4 medium buses. Cells are sometimes used to provide additional routing resource. The ability to use a core cell as a routing resource or as logic provides a programmable means of adjusting routing resource to fit design specific requirements.

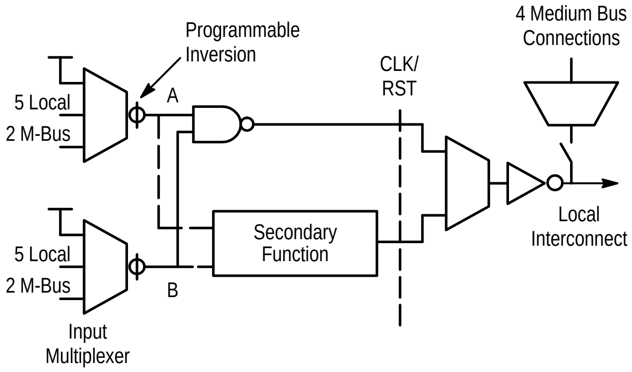


Figure 2–7. Core Cell Structure

Each cell has three states; repowering buffer, primary function, and secondary function. In addition, all cell inputs have programmable input inversion. MPA1000 core cells are organized in 2x2 groups called tiles. Within a tile, each of the 4 cells has a different secondary function (Figure 2–8). The core cell primary function is a 2 input NAND. Secondary functions include; XOR, register, and wired OR. The register element is configured as a DFF or latch with clock enable and set or reset. A special, 1ns skew, network is provided to drive register clock and reset/set pins. High performance, gate level, cells necessitate controlled clock skew to avoid negative setup time situations. The MPA cell states were chosen based on a careful analysis of macrocell utilization statistics from a large number of ASIC designs implemented in Motorola's H4C array.

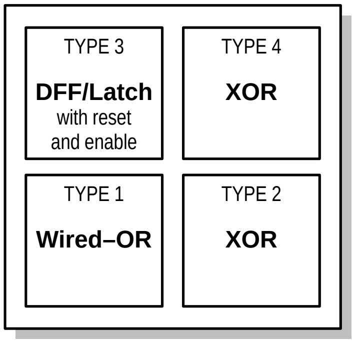


Figure 2–8. Core Cell Secondary Function

☞ One Hot State Machine Design is Preferred

Designing your state machines as one hot is usually the most efficient method for the register rich MPA.

☞ BUFF from the MICROLIB

BUFF is the only buffer available to the designer that will not get mapped out on import.

There are quite a few trivial optimizations that get made to your design during import, one of the most common is getting rid of superfluous ‘buffers’. Examples of which include INV (inverters), AN2 gates with both inputs tied together, AN2 gates with one input tied high etc. (A more complete description of this re–mapping process can be found in the on–line help of the MPA Design System under: “Help on Design → Logic Optimisation → Summary of Optimisations”.) Don’t panic at the above statement that inverters are “gotten rid of”. They are simply mapped to the correct sense on the core cell’s programmable input multiplexers. No delay penalty is incurred for inserting an INV.

☞ Tri–State drivers are not available internally

Because the MPA’s routing resources are fully buffered (actively driven) there are no internal tri–state buffers available. Designers accustomed to using such elements to allow multiple drivers access to a single data line, should instead consider using multiplexers.

☞ Wired–Or a.k.a. Open Drain

In some instances, it may be preferable to use a collection of open drain drivers to drive a single data line. The MPA library elements that accommodate this type of connection include: WINV, WOR2, WND2, and WBUF. It is important to remember that open drain drivers can only actively pull a signal low, a passive pull up resistor is required to pull the net high; that’s the job of the WPUP library element. By default, instantiating a WPUP element results in a single pull–up resistor being attached to the net. Assigning the attribute DPLD_PUP with a value of BOTH results in two pull up resistors being added in parallel to the net. The low to high transition time is thus improved, but at the expense of more static current drain when any of the attached drivers is holding the net low.

Besides lower speed, another draw back of using open drain drivers in the MPA is the restriction that all the open drain drivers within a zone must reside on the same Wired–OR Bus, and that drivers in other zones must also be in placed in the same relative horizontal position. The autolayout tool handles all of this automatically, but it does tend to reduce the number of valid solutions available to the autolayout tool for the remainder of your design.

☞ You Must Use All Macro Inputs, ONE and ZERO

The autolayout tool insists that all MACROLIB and MICROLIB inputs be used. If you don’t need a particular input for your design, you are still required to tie it to logic or a ONE or a ZERO (from the MICROLIB). There is no routing consumed when specifying a ONE or a ZERO, the tie off is made at the cell’s input selection mux. There is no fan out restriction for a ONE or a ZERO.

I/O cells

I/O cells are located at the device periphery surrounding the quadrants (Figure 2–4 on page 2–4). Besides direct input and output, each I/O cell can be configured to be; input, output, bidirectional, registered input, registered output, registered I/O. The two registers can be independently configured as a latch or D–type flip flop. Input register setup time is adjustable to compensate for clock network input delay. Input buffer threshold adjustment provides either TTL or CMOS levels. Output buffer drive capability is programmable to 4mA or 6mA. And each output can be independently programmed to either 3V or 5V levels with slew rate control. The output buffer can be configured as an open drain to facilitate system level wired OR applications. Figure 2–10 sums up I/O cell structure. Dedicated, fully IEEE 1149.1 compliant boundary scan is also provided.

The output buffers of unused I/O cell outputs are “turned–off” presenting a high impedance load to the external world. Similarly, input buffers of unused I/O cell inputs are also “turned–off”; there is no requirement to tie unused inputs high or low.

The MPA’s output drivers are actually composed of a pair of 4mA drivers, only the second of which has controllable slew rate.

Default output configuration is the first 4mA driver. If the user attributes the I/O cell (or its formal port) with DPLD_OPDRIVE set to a value of 6mA, then the second 4mA driver will be added in parallel.

If the design calls for multiple 6mA drivers to be switched simultaneously, the designer should consider also attributing the outputs with DLPD_OPSLEW set to a value of “low”. Doing so decreases the di/dt term in the familiar V = L di/dt equation, thus reducing ground bounce.

```
instance outbuff attribute dpld_opdrive 6ma
instance outbuff attribute dpld_opslew low
```

Figure 2–9. Sample .PAT Entries for a 6mA, Low Slew Rate Output Called “Outbuff”

Table 2–3. Slew Rates for the MPA1000 Family (Note 1.)

Output Conditions	t_r (ns) at 5V	t_f (ns) at 5V	t_r (ns) at 3.6V	t_f (ns) at 3.6V
DPLD_OPDRIVE=6mA & DLPD_OPSLEW=high	1.7	2.0	0.9	1.2
DPLD_OPDRIVE=6mA & DLPD_OPSLEW=low	0.6	1.0	0.3	0.9
DPLD_OPDRIVE=4mA & DLPD_OPSLEW=high (Note 2.)	1.1	1.4	0.6	1.0

1. Measurements taken between 10% and 90% of V_{DD} at 25°C, C_L = 50pF. Note that DPLD_OPDRIVE = 4mA with DPLD_OPSLEW = low is an illegal combination.
2. Default values.

☞ Start Off Easy, Begin with IPBUF, OPBUF, IPCLK, IPRST

The Complex I/O can be a space and a time saver for your more critical designs, but you may want to consider starting off slow and use the simpler I/O structures.

☞ Enable and Reset Pins on Complex I/Os do not have to be tied

There are too many permutations possible in the I/O cell to make each available as a unique macrocell in the IOLIB. Consequently a short cut has been made available to the designer using Complex I/O, namely it is not necessary to tie reset or clock enable inputs high or low when using elements of the IOLIB. (N.B. This is not true for elements from the MICROLIB or MACROLIB. Each of these inputs must be used or otherwise tied off.) The autolayout software will make the obvious assumptions about how the unused input should be tied and make the tie off for you.

☞ Don't fix your I/O locations unnecessarily

Fixing your I/O locations using DPLD_PAD_PLACE attribute may place an undue burden on the autolayout tool. Most designs will route to a higher performance level if the autolayout tool is given as much freedom as possible with regards to I/O pin placement.

☞ Twinning Outputs

Two outputs can be connected in parallel to increase the the output drive current, however, to avoid contention between drivers, care must be taken to insure that output signals are synchronized. Use the following as guidelines:

- Connected outputs must reside in the same I/O zone. (The I/O pad ring is divided into zones each containing 5 I/O cells and two primary clocks. To identify a zone in a packaged product, look for groups of 5 adjacent I/Os in the pinout assignment.)
- Output signals must be gated through the I/O flip–flop registers.
- Output flip–flops must be clocked by a common primary clock signal via the clock distribution network, which is balanced and has a skew of <1ns between any two registered clock inputs. (Primary clock signals are the only way in which the I/O flip–flop clocks. Clock signals may originate external to the device via library element IPCLK or from the array by routing the signal to the primary clock bus via APCLK.)
- To reduce ground bounce, twinned outputs should be as close as possible to a VSSE pin. If ground bounce persists, alternate slew rate – fast on one, slow on the other.
- Using open drain output is a safer alternative, although, the speed will be limited by the pull–up resistor.

Hierarchical Routing Resources

The MPA interconnection structure is partitioned into 3 levels; Global, Zonal (or medium), and Local. Local interconnection is used to connect a core cell to 8 of it's perpendicular neighbors (Figure 2–11). Zonal interconnect consists of the medium buses and connects groups of cells within a zone (Figure 2–12). Global interconnect includes global buses, x buses and interquadrant switches (Figure 2–4 on page 2–4). Global buses provide quadrant and chip level inter–zone and zone to I/O cell interconnections. Special interconnection resources are also present and consist of clock distribution, wired–OR and peripheral bus. Routing specialization provides an opportunity for level specific performance optimization. Specialization also diminishes the amount of interconnection options required at each core cell, reducing cell size and boosting silicon efficiency.

Local Interconnect

Local interconnect provides the fastest path between 8 neighboring core cells. Local interconnect is continuous across the device and is not effected by zonal boundaries. Local interconnection favors frequently used connections, the cell to the immediate left and immediate right of the driving cell have 2 connections. Local connections are used for high performance intrazone connections and are also used to cross zone boundaries when necessary.

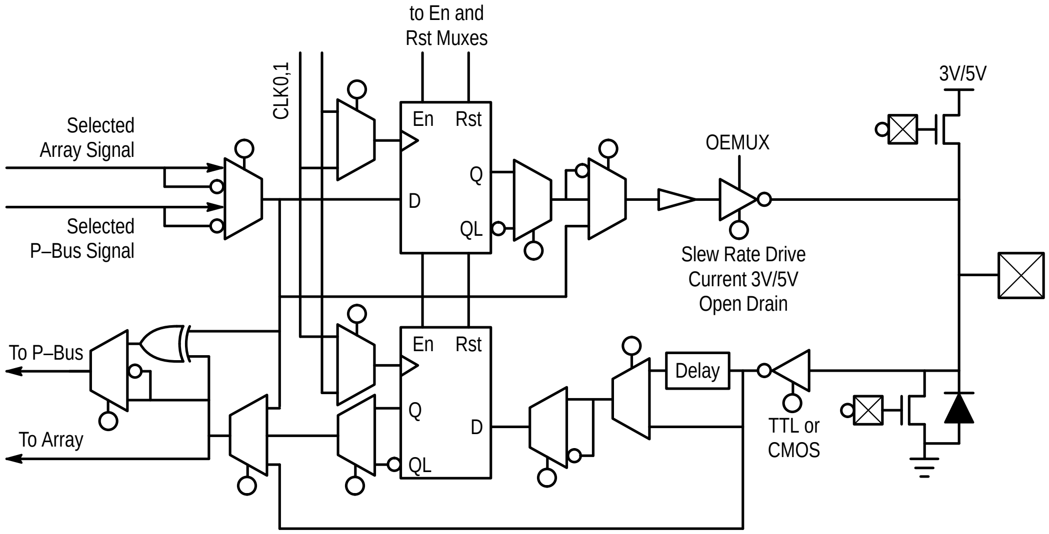


Figure 2–10. Input/Output Cell Structure

Medium Interconnect

Medium interconnect spans a single zone and provides intrazone connections beyond the span of local interconnect or for connection of zone cells to global signals through the port cells. There are 4 horizontal and 4 vertical medium buses per core cell. Medium bus connectivity to core cells is sparse to minimize loading and limit core cell input multiplexer size. This connectivity is arranged so that a tile can be fully connected to the 16 medium buses which cross it.

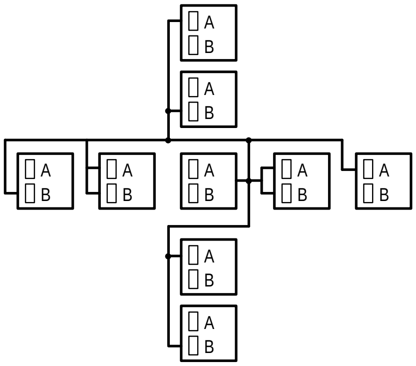


Figure 2–11. Local Interconnect

Port Cells

At zone edges, port cells provide a bridge between global resources and zonal resources. Port cells transport signals into and out of a zone and are the only interface between zonal and global resources (Figure 2–13). All 4 medium buses, 4 global buses and the x bus in a given row or column connect to the port cell.

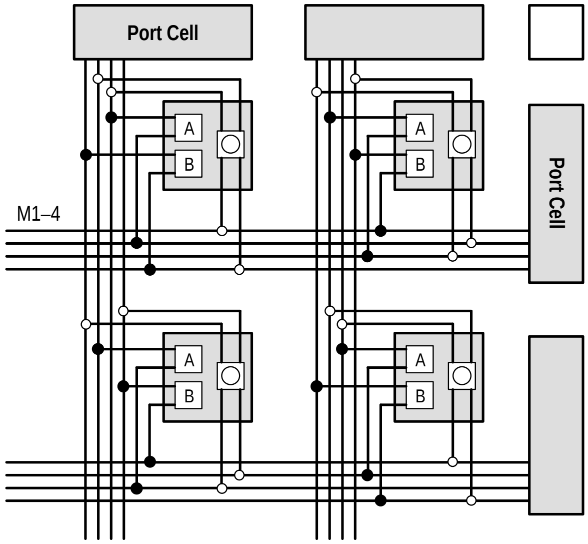


Figure 2–12. Medium Interconnect

Port cells also provide connections to 4 of the 8 low skew clock distribution lines which span the device. Port cells also provide global to x bus access and serve as a pathway for zonal wired OR buses to connect to global busses.

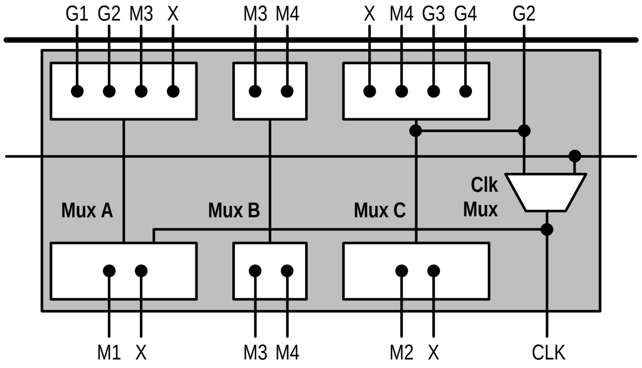


Figure 2–13. Vertical Port Cells

Global Interconnect

Global interconnect consists of global buses, x buses, interquadrant switches. There are 4 horizontal and 4 vertical global buses passing over each core cell. All Global buses only connect to the port cells, I/O cells and interquadrant switches. Global buses span a quadrant and are used to interconnect the zones within the quadrant together. Between quadrants, interquadrant switches connect two global buses together to form a device level connection.

Each core cell contains a x–bus switch (Figure 2–16) which is independent of cell logic or interconnect functions. A single vertical and a single horizontal x bus passes over each core cell and connects to this switch. Each x bus connects to all the core cells is a single zone column or row and terminates at the port cells on opposite edges of the zone. Each x bus has 10 connections inside the zone and 2 port cell connections. Port cell connections are used to make x to global, x to x and medium to x connections. Medium to x connections are used to hop over a single zone X buses are used to facilitate 90° global bus turns and provide a means for global bus fanout.

High Fan Out

As mentioned previously, the routing resources of the MPA are fully buffered. There is no reason for the designer to concern himself with loading effects of high fan out net. However, high fan out nets can have an undesirable impact on routing resource consumption. Using only local routing, a single driver could under the most ideal conditions drive only 8 local neighbors. In real world designs however, each of the destinations of a high fan out net has its own downstream circuitry associated with it; there is a vanishingly low probability that they will be placed in the 8 local adjacent locations. For fan outs greater than 8, exclusive local routing is impossible, and both medium and global routes will be used to complete the net. If the fan out is large enough, and the circuitry placed sufficiently far apart in the array, routing resource consumption may become problematic.

The primary clock and reset distribution network may be used to route high fan out signals. Driving the high fan out net internally with an ACLK or ARST buffer, or externally with an IPCLK or IPRST buffer will put the signal on one of the 8 global Clk/Rst distribution lines. The routing congestion can thus be solved, but at the expense of reducing the clock and reset routing solution space. Do not route nets to I/O (other than Clk/Rst) on the primary clock network. There is no mechanism for completing such a route on the MPA devices.

For software versions 2.4 and later, ACLK and ARST insertions for high fanout nets will be automatic.

Delays in Routing

Both PCB and older ASIC designers share the mind set that delay through a multi–level logic path is principally a function of “gate delay”. In the ASIC world, routing paths are as short as possible and do not pass through multiple levels of pass gates, muxes, and buffers. Similarly, a PCB trace is a simple and hopefully short run of metal, with most of the “gate” delay happening as a function of package input and output delays. A “logical” net in an FPGA however may be a series of several different electrical nodes, each being separated by a mux or switch of some type. The consequence of this is that “routing delays” not gate delays are the first order factor determining the resultant circuit’s speed.

Empirical analysis of several hundred sample designs suggests that a multiplication factor of 2.4 can by applied to the sum of a path’s gate delays to come up with a very rough estimate of what the post autolayout total path delay might be. There are many factors that influence that actual number, so please consider this only as a very crude estimate.

S–R Flops, Avoid the Temptation

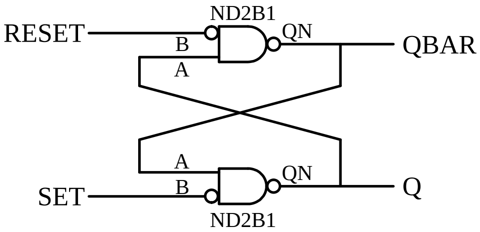


Figure 2–14. A Classic S–R Flop;
An Accident Waiting to Happen

The above construction of an asynchronous S–R flip flop is familiar to all, but should be so for its unfavorable characteristics. Remember that routing delay in an FPGA is the highest order term in delay equation. In the above construction, the (active high) SET pulse width must be greater than the ND2B1 propagation delay plus Q to A routing plus another ND2B1 delay plus QBAR to A routing delay. Without a detailed analysis of the post autolayout path delays, the pulse width specification can not be known. The same holds for the RESET pulse width. A new autolayout run on the same design may alter these path lengths considerably. Additionally this sort of asynchronous feedback loop will generally cause back annotation, simulation and timing analysis tools trouble.

Avoid asynchronous design.

☞ Delay Lines, Avoid the Temptation

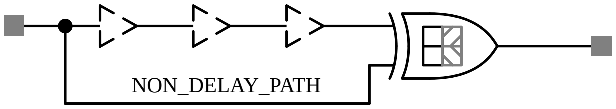


Figure 2–15. A Delay Line for Turning Edges into Pulses, a Dangerous Proposal

Remember that in an FPGA routing is not just a piece of wire. Routing is comprised of wire, muxes and pass gates. In the above example, the intent is to turn a rising or falling input edge into an output pulse. The assumption is that the “NON_DELAY_PATH” will have a shorter delay than the “delay line” formed by the series of BUFF elements. Again, the MPA design software does not guarantee minimum delays and so it is possible that the an autolayout run might result in the NON_DELAY_PATH to have a delay significantly close the delay line path. The circuit may not work.

Avoid any design habit that makes assumptions about minimum delays, even for just plain routes.

I/O Cell Connections and Peripheral Bus

I/O cells are a pathway between array and bonding pads. Global buses, x buses and adjacent zone medium buses can be connected to I/O cells at quadrant edges. Each I/O cells is directly connected to the adjacent bonding pad.

A specialized bus, called the peripheral bus, resides in the I/O cell – quadrant interface (Figure 2–4 on page 2–4). The peripheral bus comprises 8 lines which are interrupted at device corners by a peripheral bus switch similar to the interquadrant switch. This switch joins peripheral bus segments to create connections spanning more than a single device edge. Peripheral buses carry I/O control signals common to two or more I/O cells such as a latch enable or tristate control signal. The I/O cells can also drive these buses with an open drain device. When combined with programmable pullups located in the corners of the device, the peripheral bus can be used to form wide gates for address decoding (Figure 2–17).

☞ Use the P–Bus to route enable signals

Whenever an enable signal goes to more than one I/O cell, it is recommended that the designer employ the P–Bus (by inserting and APBUF).

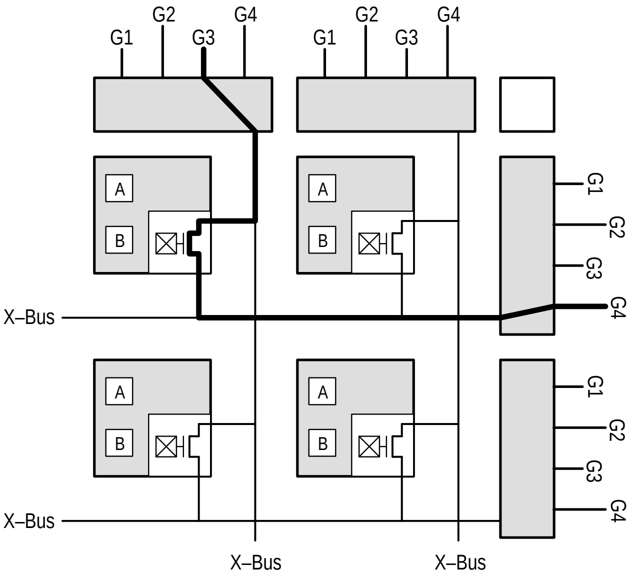


Figure 2–16. Global Bus Turn Using the X–Bus

Wired OR Nets

Wired OR nets are constructed using type 1 core cells. When the type 1 secondary function is enabled, the NAND drives an open drain device directly connected to a special bus shared by all the type 1 cells in the same zone row. This bus, the zone wired OR bus, terminates in the port cell and has a single, dedicated, pullup. When this bus is used, the port cell wired OR to global bus connection and the global bus pullup located near the interquadrant switch are enabled. These resources can be used to map 3–state buses onto the MPA1000 device.

Clock Distribution

Clock distribution is implemented through a dedicated, low skew, network consisting of; 8 dedicated clock input lines connected to 2 I/O cells on each device edge, a central clock buffer, a distribution comb structure, zone corner clock selection cells and the zone port cells along the top of each zone. The zone corner cell selects 2 of the 8 lines for zone clocks and 2 of the 8 lines for zone reset (Figure 2–18). Zone registers are connected to these clock and reset signals through the top row of port cells. The comb extends into the I/O cells via a similar clock selection cell attached to each group of 5 I/O cells. This group is called an I/O zone. All 8 clock lines can be driven from the I/O bonding pad or the array. The distribution network is balanced and has a skew of < 1ns between any two register clock inputs.

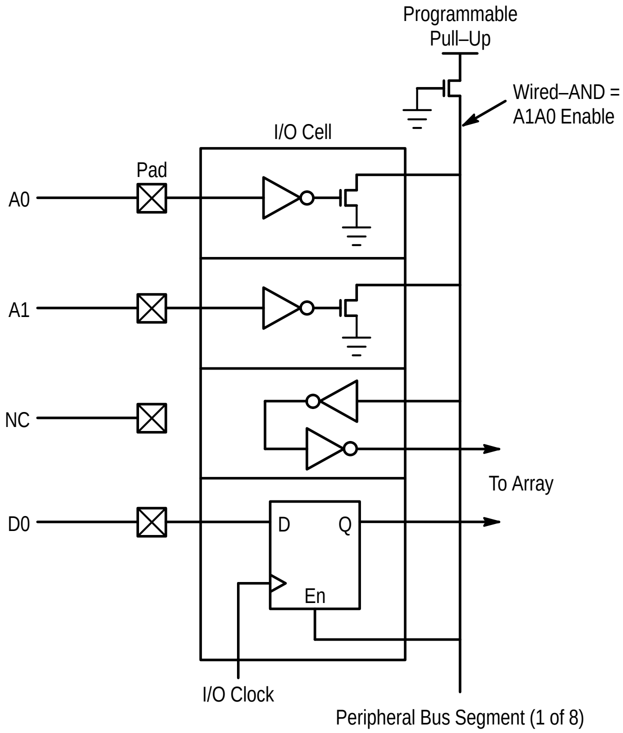


Figure 2–17. Using the Peripheral Bus for Address Decoding

Connections in the port cells allow the clock network to drive zone logic in addition to the register clock and reset inputs. Unused clock lines can be used for efficient distribution of any high fanout signal. If there are more clocks in the design than clock resources, the MPA design system automatically constructs a comb from global buses to generate a secondary clock network with a skew of < 3ns. Secondary clock construction is facilitated by a port cell connection which provides non–clock network access to zone register clock and reset inputs.

☞ **Secondary Clock Networks Consume Routing Resources**

The MPA easily handles a fair number of secondary clock networks, but networks with large numbers of Clk/Rst loads are more efficiently accommodated by moving onto the primary Clock Distribution Network using ACLK or ARST buffers mentioned previously.

Tertiary clock (reset) networks are identified by the autolayout software as any net driving four or fewer clock (reset) inputs not on the primary clock distribution network. There is no skew guarantee on these tertiary clock nets; they are routed on normal resources.

☞ **Too Many Clocks**

The MPA is best suited for designs with a few primary clocks, but multiple clocks are supported. The problem with tertiary and especially secondary clock networks is that they consume a fair amount of routing resources. An otherwise easy to fit design may not be routable once multiple secondary clocks are accommodated.

☞ **Gated Clocks, Avoid When Possible**

Inserting anything but an INV in a clock path will result in the clock being pulled off the primary clock network and placed either secondary or tertiary routing (depending on the number of clock loads downstream of the inserted gate). As mentioned above, this tends to spread the resulting layout out a bit more and consequently can slow things down some. If a gated clock is desired, try instead using register elements with clock enables.

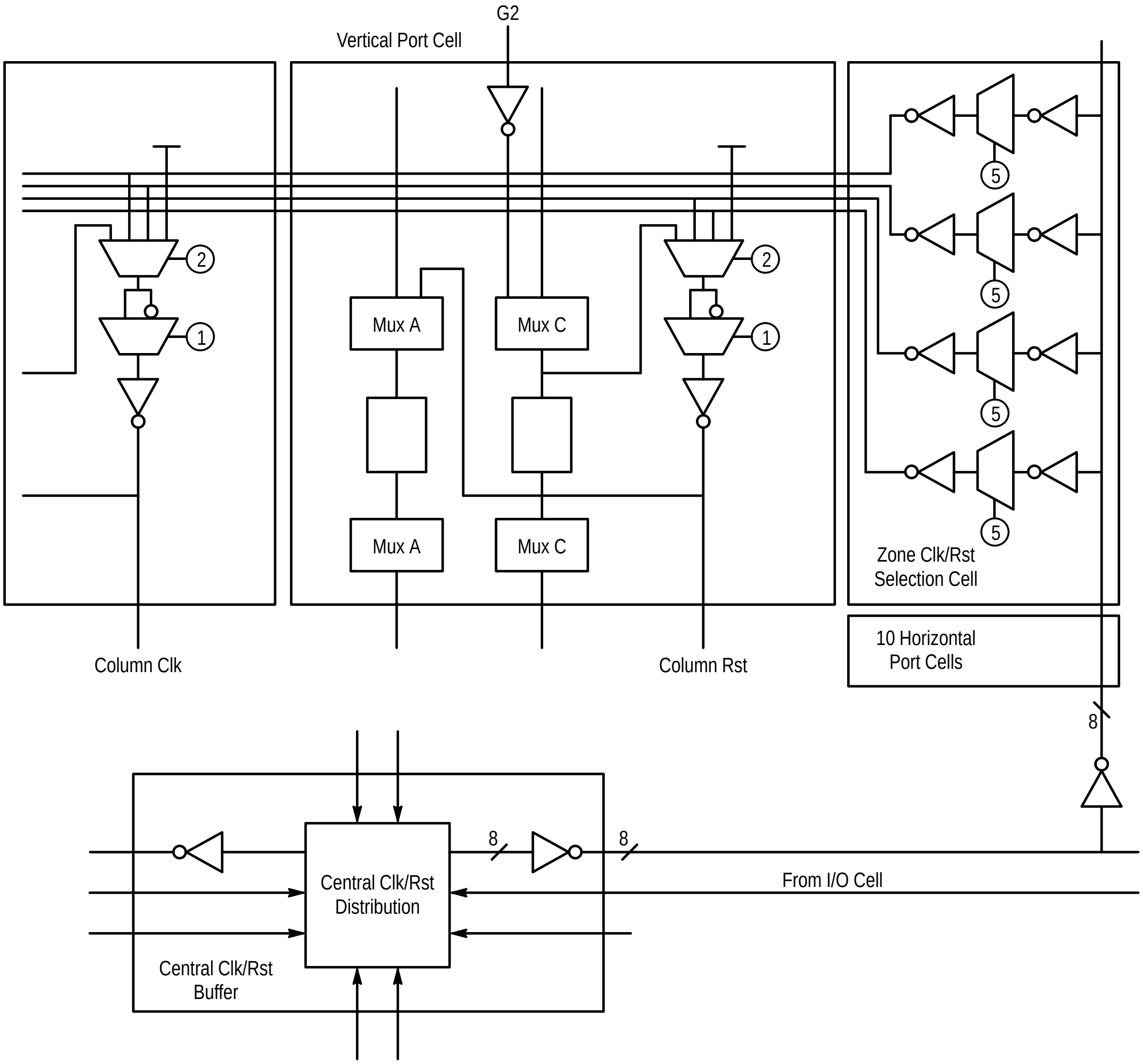


Figure 2–18. Clock Distribution Network Connectivity

☞ ACLK & ARST Consume Clk/Rst I/O Sites

Each ACLK and ARST buffer used resides in one of the 8 clock pad locations. Using an ACKL or ARST consumes this pad location such that it is no longer available to use as an I/O site. The designer is allowed a total of 8 ACLK, ARST, IPCLK, IPRST cells in his design.

☞ I/O Cells Can Only Be Clocked From the Primary Clock Distribution Network

Clocking I/O macros via secondary or tertiary clocks is prohibited. Reset is however permitted to be sourced from the array or Peripheral Bus (P–Bus).

☞ Clock Sense Selection is Made in the Vertical Port Cell

All flops within a column will have the same clock and reset (or will lie unused).

☞ Do Not Use the Primary Clk/Rst Distribution Network to Route Clock Enable Signals

Referring to Figure 2–18 on page 2–12, note that a clock is paired with a reset and brought down to all 5 of the Type 3 cells within a column. If the associated clock enable (if used) is also on the primary clock network, there would be no efficient route available to get it down to the target flops. Do not use the Primary Clock Distribution Network to route clock enables. (Do use it for “Latch Enable” signals.)

MPA1000 Device Configuration

Configuration Overview

MPA1000 devices have an SRAM configuration memory. Configuration memory contents completely define MPA device function. The MPA1000 design system generates configurations from completed layouts. On chip control logic loads configurations in one of four modes automatically on power up or under external control. MPA1000 devices have a very rapid configuration load cycle, infinite reload and are in system reconfigurable. The configuration modes are; Boot From ROM (BFR1:3) and microprocessor peripheral or MICRO Mode. In either mode, multiple devices can be daisy chained to form a large programmable subsystem.

In all BFR modes, the MPA device controls configuration and loads from either a byte wide or serial memory. In BFR mode 1 (Figure 2–20), the device generates 18 bits of address and reads 8 bits of configuration data. MPA devices generate 18 bits of address or 262K bytes (e.g., 17 MPA1036 devices). If a larger address range is required, BFR mode 3 (Figure 2–26 on page 2–20) can be used. In BFR mode 3 an external address generator is used to extend the address space. BFR mode 2 is a special case of BFR 3. In this case the address generator is resident in the external serial EPROM and data is presented to the device 1 bit at a time. The MPA design system download POD and the MPA17000 serial EPROMs are used with this mode.

In MICRO Mode, the MPA1000 device becomes an 8 bit peripheral slave device. A microcontroller or microprocessor controls the configuration process. MICRO Mode provides more control over configuration and user mode device behavior than other modes. For example MICRO Mode can be used to both write and read configuration memory.

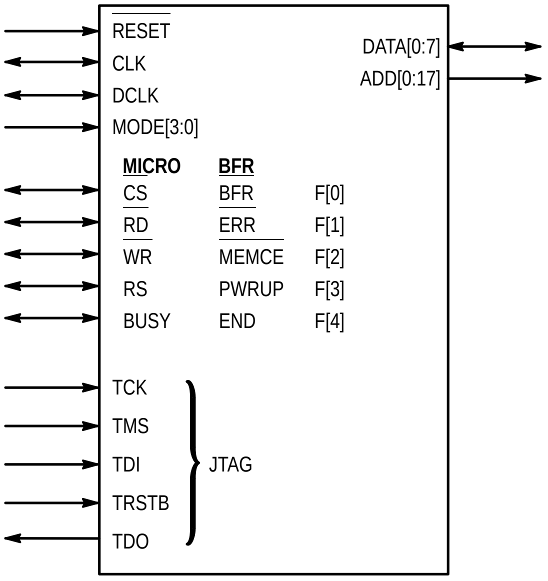


Figure 2–19. Configuration Interface Signals

Configuration information generated by the MPA Design System includes Error Check Bytes (ECBs). ECBs are used to detect configuration data corruption while configurations are loaded into the device. The configuration process halts and error status is indicated if an ECB mismatch is detected anytime during the configuration process. ECB checks insure the integrity of configuration data and protect MPA devices from possible damage.

Depending on the selected configuration mode, some user I/O pins become unavailable for post configuration use. These pins are listed as “dedicated” in Table 2–7 on page 2–24 and Table 2–5 on page 2–16. The system level interface for MPA configuration is shown in Figure 2–19. Note that the meaning of the F[4:0] pins is mode specific, refer to Table 2–7 and Table 2–5 for detailed signal descriptions.

Table 2–4. MODE[3:0] Pin Programming

Mode Bits				Description
[3]	[2]	[1]	[0]	
X	X	0	0	**MICRO Mode** — Micro–processor/controller interface circuitry with parallel (byte wide) data.
X	X	0	1	**BFR Mode (1)** — Boot From ROM, byte wide data. MPA generates ROM addresses.
X	X	1	0	**BFR Mode (2)** — Boot From ROM, serial data. (Low pin count serial EPROM generates own addresses.)
X	X	1	1	**BFR Mode (3)** — Boot From ROM, byte wide data. MPA does not generate ROM addresses.
X	1	X	X	Use external clock for configuration.
1	X	X	X	Enable JTAG circuitry and pins.

Configuration Clock

The MPA1000 device has an internal oscillator. The internal configuration clock is derived from the oscillator and is presented at the CLK pin when MODE[2] is low. When MODE[2] is high, the internal clock is disconnected from the oscillator and an external clock must be presented on the CLK pin to configure the device.

The configuration clock drives the configuration logic and its associated state machine. If using an external configuration clock, it is necessary to provide it always to ensure RESET, BFR and PWRUP signal transitions are detected and handled in the expected fashion by the configuration logic.

Bootstrap Voltage

Signal pathways in the MPA 1000 device are controlled with n–channel transistors. The gates of these transistors are connected to individual SRAM configuration memory cells. To pass a rail to rail signal through these transistors during user operation, the gate voltage must be elevated above V_{DD}

to compensate for transistor threshold and body effect voltage drops. MPA1000 devices contain a charge pump to generate this elevated voltage, called the bootstrap voltage. The charge pump is connected to the supply line of each SRAM cell and is driven by the internal oscillator.

Since configuration memory is generally not dynamically changing during user operation, the charge pump must only supply small leakage current losses and is not designed to supply sufficient current for SRAM write operations. During configuration, the charge pump (bootstrap) is internally disabled by shunting the SRAM supply to V_{DD} through a large p–channel device. In order for the charge pump to operate properly, the internal oscillator as well as the bootstrap circuitry must be enabled. In MICRO Mode, the processor has control over these functions. In BFR modes, the on chip configuration controller insures proper sequencing of these controls.

The MPA1000 device is only guaranteed to function properly with bootstrap enabled. The internal oscillator must be running and bootstrap should be activated 100µs before user inputs or outputs are enabled. If dynamic configuration modification is desired, the bootstrap voltage can be supplied externally on the V_{PP} pin and MICRO Mode can be used to disable bootstrap by shutting off the on–board oscillator. The bootstrap voltage should be V_{DD} + 1.5V. At no time should V_{PP} exceed 6.5V.

JTAG

The MPA1000 device contains dedicated JTAG IEEE 1149.1 boundary scan circuitry. JTAG can be used on configured devices. JTAG is enabled any time MODE[3] is raised. When MODE[3] is high, 5 user I/O pins become JTAG controls and user mode operation of those pins is interrupted. Since the TAP controller can take control of all device pins, care must be used to prevent the TAP controller from interfering with device user mode or configuration operation.

Boot From ROM (BFR) Modes

In BFR modes, the MPA device controls device configuration and assumes a memory–processor interface to the configuration store. The MPA device either asserts addresses directly (internal address generation) or issues address reset and increment pulses (external address generation). Data is read either serially or 8 bits at a time. Table 2–5 describes BFR interface signal operation. ADD[17:0] are only used in BFR mode 1. DATA[7:1] are not used in BFR mode 2 (serial data).

A BFR load sequence is initiated by; a falling edge of $\overline{\text{BFR}}$, device power up or a rising edge of RESET. MEMCE falls to indicate the start of a configuration load sequence. On subsequent alternate rising edges of CLK, the data bus value is latched. The configuration process terminates when a complete configuration is successfully loaded and END is asserted or when a configuration error is detected and ERR is asserted. After END is asserted, the device will begin user mode operation 3 clocks after PWRUP is asserted or 3 clocks after END if PWRUP was already high. All configuration timing is synchronous with the internal or externally supplied configuration clock. Figure 2–22 describes BFR sequence timing details.

All BFR sequences begin with an internal device reset sequence where the entire configuration memory is reset. The duration of this sequence depends on the size of the MPA device being configured. A falling MEMCE edge indicates configuration commencement and data loads begin after 2 subsequent configuration clocks. The first positive edge of DCLK signals the external address generator to increment the byte or bit address. Prior to MEMCE assertion, DCLK is tristated.

The duration of the configuration process is also dependent on device size. Configuration duration can be estimated for BFR 1,3 by dividing the total number of configuration bytes by 1/2 the configuration clock frequency. For example, the MPA1036 device has 139 rows of 105 bytes including the ECB or 14,595 bytes. If the configuration clock is 2MHz, configuration will take approximately 15ms. If BFR 2 is used, the configuration process will take approximately 8 times longer. See “Device Configuration Memory Organization” on page 2–32 for device specific configuration memory sizes.

Table 2–5. BFR Mode Configuration Control Pins

Pin Name	BFR	I/O		Description
MODE[3:0]	MODE[3:0]	I	Dedicated*	**Configuration mode**
RESET	RESET	I	Dedicated	**Configuration reset** — Clear configuration memory. Configure when released.
CLK	CLK	I/O	Dedicated	**Configuration clock** — If MODE[2] is low, the internal configuration clock is presented. If MODE[2] is high, an external clock must be supplied.
F0	BFR	I	Dedicated	**BFR initiate** — A falling edge starts a reset and configure sequence.
F1	ERR	O	Dedicated	**Error** — Configuration checksum (ECB) or incorrect device ID error. **Open drain output**
F2	MEMCE	O	Dedicated	**Memory Enable** — Active low during configuration sequence.
F3	PWRUP	I	Dedicated	**Power up** — After configuration complete; enable bootstrap, enable user inputs, enable user outputs. Often simply tied to V_{DD}.
F4	END	O	Dedicated	**Configuration completed** — Asserted when a configuration has been successfully loaded into the device.
DCLK	DCLK	I/O	Dedicated	**Data clock** — Each output pulse indicates current data bus value has been latched and data address should increment. Becomes an input after configuration completes.
DATA[7:0]	DATA[7:0]	I	User/Data	**Data port**
ADD[17:0]	ADD[17:0]	O	User/Address	**Address output** — If internal address generation is selected. (BFR Mode 1)
JTAG[4:0]		I/O	User/JTAG	**JTAG pins** — Active when MODE[3] is asserted.

* Dedicated — Pins used for configuration. Not available for user I/O.

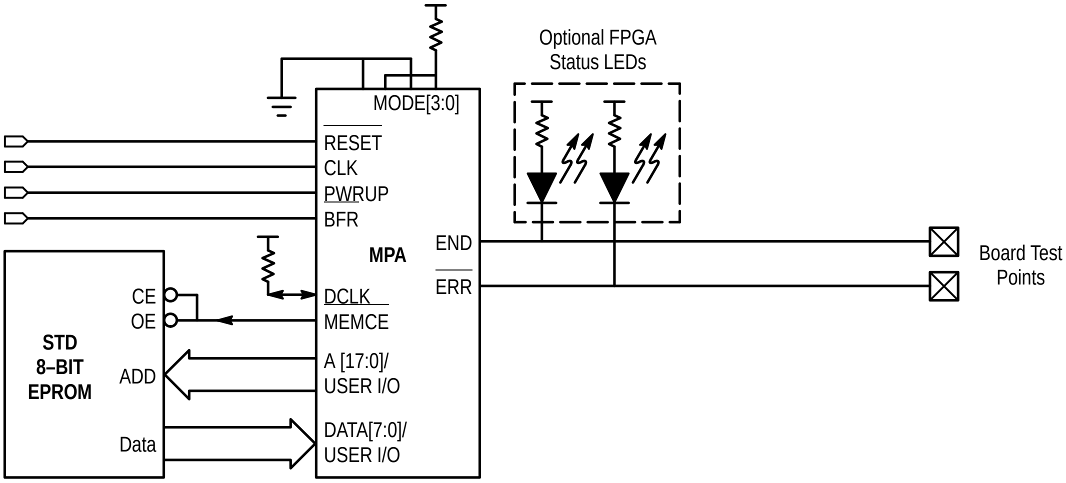
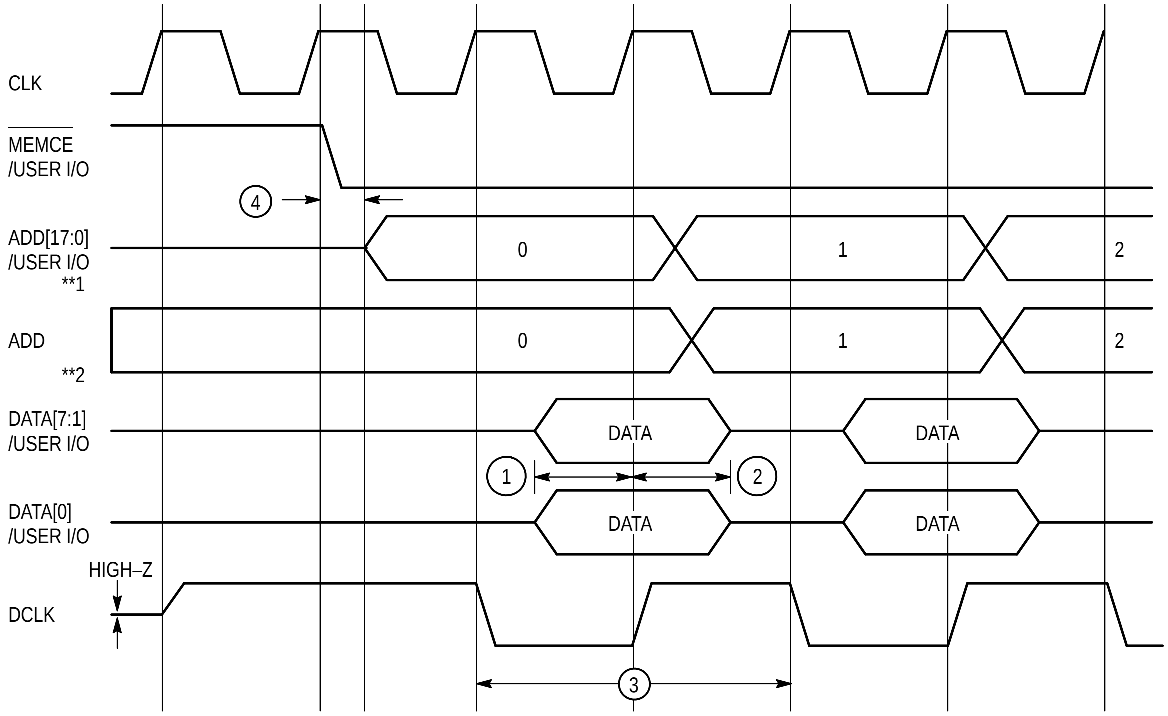


Figure 2–20. BFR Mode 1: 8–Bit Data, Internal Address, External Clock

BFR Mode 1 Operation: 8 bit data, Internal Address Generation

In BFR 1, MPA configuration logic asserts an 18 bit address and reads data 8 bits at a time as shown in Figure 2–20 on page 2–16. A paging scheme could also be used where additional upper address bits were provided by an external page register. Multiple configurations could be accessed by writing the page register, asserting BFR, and self loading the referenced configuration.

ADD[17:0] are tristated during device reset, asserted during configuration and released for user mode operation. DCLK is tristated until 1 clock prior to MEMCE assertion. The first address is asserted coincident with the falling edge of MEMCE and the data bus is latched 2 configuration clocks later. The internal address counter is incremented on each positive DCLK edge (Figure 2–21). This process proceeds until an entire row of configuration data is loaded into the internal row data register and the ECB is verified. ADD[17:0] (current address) and DCLK (=1) hold while the internal write cycle takes place. Start Access (SA) marks the beginning of the write cycle and End Access (EA) marks write completion (Figure 2–27). After the write completes, the address presentation and data latching process resumes. When the entire device configuration is loaded, END is asserted, DCLK is tri–stated and 2 clocks later user inputs are enabled and MEMCE is deasserted. One additional clock and user outputs are enabled and user mode operation commences. If the written ECB does not match the internally calculated value, ERR is asserted 2 clocks after the ECB is written. Once ERR is asserted, the configuration process halts and cannot be restarted until a new configuration process is initiated using BFR, RESET or a power down. When END is asserted, DCLK becomes an input and the internal address counter remains active until PWRUP is asserted. Figure 2–30 shows how this can be used in a multiple device subsystem. Because DCLK becomes an input, it must be tied high with a weak pullup when used in a single device configuration (Figure 2–20) to prevent a floating input condition.

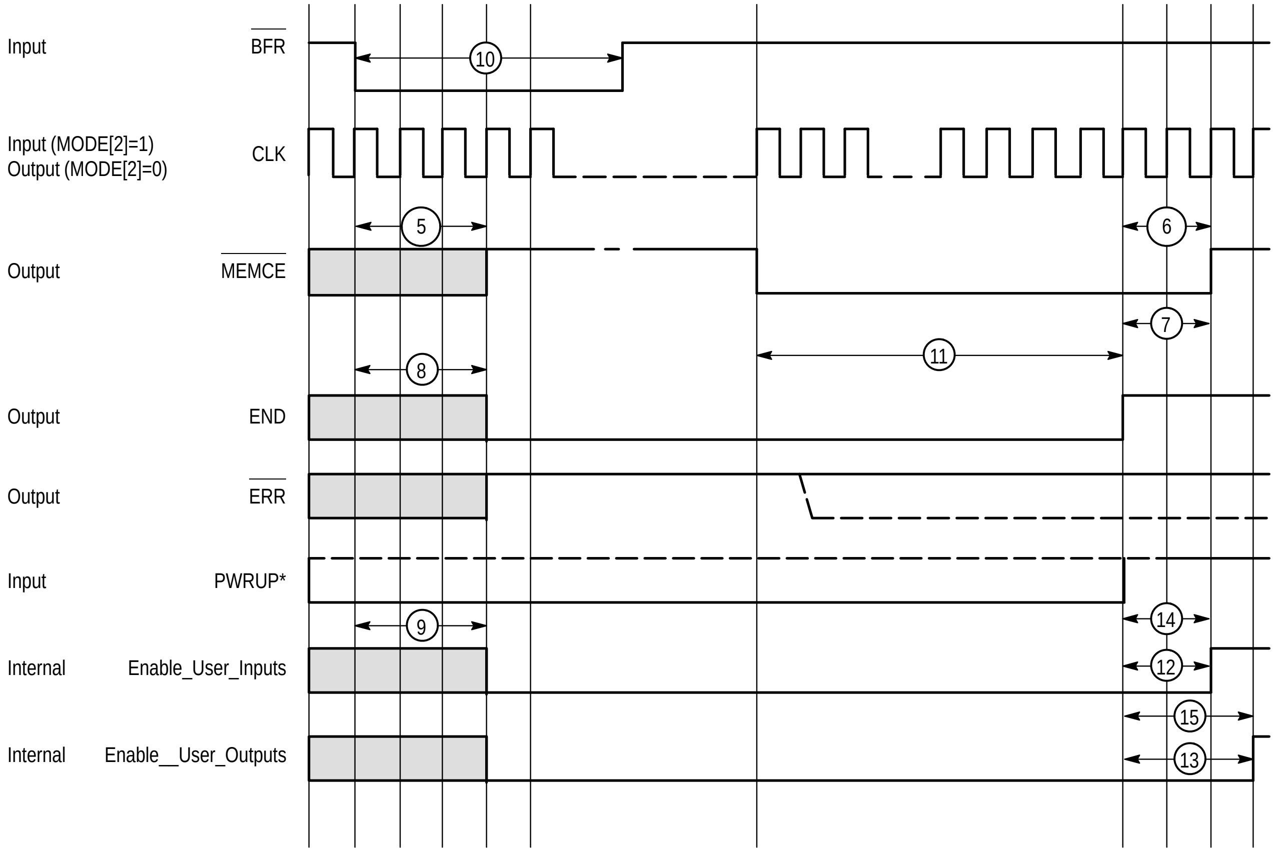


**1 – Internally generated Address

**2 – Externally generated Address

Figure 2–21. BFR Data Access Detail

Number	Characteristic	Min	Max	Unit	Notes
1	Data Setup to DCLK	20		ns	
2	Data Hold after DCLK	0		ns	
3	DCLK Period (When Active)	2	2	CLK	
4	CLK to Address Valid (Internal Generator)	15		ns	



* PWRUP can be, and usually is, tied to V_{DD} internally.

Figure 2–22. BFR Sequence

Number	Characteristic	Period	Unit	Notes
5	$\overline{\text{BFR}}$ Low to $\overline{\text{MEMCE}}$ High	3	CLK	If $\overline{\text{BFR}}$ reasserts durning a boot
6	END High to $\overline{\text{MEMCE}}$ High	2	CLK	
7	PWRUP to $\overline{\text{MEMCE}}$ High	2	CLK	
8	$\overline{\text{BFR}}$ Low to END Low	3	CLK	Note 3.
9	$\overline{\text{BFR}}$ Low to Internal Disable	3	CLK	Note 3.
10	$\overline{\text{BFR}}$ Pulse Width	50	ns	Minimum
11	Configuration Sequence Duration			Configuration sequence dependent on device size
12	END to Enable User Inputs	2	CLK	If PWRUP asserted, Note 4.
13	END to Enable User Outputs	3	CLK	If PWRUP asserted, Note 4.
14	PWRUP to Enable User Inputs	2	CLK	Note 5.
15	PWRUP to Enable User Outputs	3	CLK	Note 5.

3. $\overline{\text{BFR}}$ is usually an asynchronous input, 4 CLKs assumes T_{SO_BFR} is met.
4. PWRUP can be, and usually is, tied to V_{DD}.
5. PWRUP may be an asynchronous signal, 2,3 CLK, assumes T_{SO_PWRUP} is met.

A Sample BFR Mode 2 Load Sequence

The most common boot configuration for the MPA is the BFR Mode 2, using a serial boot (E)EPROM. The timing overview for such a boot load is given in Figure 2–23 and Figure 2–24, with timing notes in Table 2–6.

In this example the CLK signal can either be sourced by the MPA or generated externally and received by the MPA (according to the state of the MODE[2] pin). BFR is usually asynchronous, Figure 2–23 assumes the falling edge of BFR meets the set–up requirement with respect to the rising edge of the CLK signal. Three CLKs later The END signal de–asserts and a reset sequence begins. The length of the reset sequence is a function of the array type as shown in Table 2–6. As the reset sequence ends DCLK (connected to the EPROMS clock input) goes high, then MEMCE asserts (connected to me EPROM's RESET/OE pin. The first bit of configuration data will appear at the EPROM's data pin after this falling MEMCE. Data is latched into the MPA as DCLK is raised. The next rising edge of DCLK causes the EPROM to shift out the second configuration bit, and so on.

The internal configuration SRAM of the MPA is loaded up one row at a time. The number and width of the rows varies by array type. After a row's worth of data is read in to a configuration shift register, the MPA holds DCLK high for 12

Table 2–6. BFR Mode 1 Sequence Timing for All MPA Family Members

Number	Characteristics	CLKs	Notes
1	BFR Low to MEMCE Low MPA1016 MPA1036 MPA1064 MPA1100	 971 1411 1851 2291	Internal SRAM Reset Sequence =21+(10*95), 95 SRAM Rows =21+(10*139), 139 SRAM Rows =21+(10*183), 183 SRAM Rows =21+(10*227), 227 SRAM Rows
2	Low to DCLK Hold Off MPA1016 MPA1036 MPA1064 MPA1100	 1232 1760 2288 2800	Shifting in ID and first row of SRAM data =80+(2*576), ID & data type then 576 bits/row =80+(2*840), ID & data type then 840 bits/row =80+(2*1104), ID & data type then 1104 bits/row =80+(2*1360), ID & data type then 1360 bits/row
3	Internal SRAM Row Load	12	All devices, every row
4	Subsequent Row Sequence MPA1016 MPA1036 MPA1064 MPA1100	 1163 1691 2219 2731	Shifting in row data =12+(2*576)–1, 576 bits / row =12+(2*840)–1, 840 bits / row =12+(2*1104)–1, 1104 bits / row =12+(2*1360)–1, 1360 bits / row
	BFR Low to User Outputs Enabled MPA1016 MPA1036 MPA1064 MPA1100	 111,540 236,544 408,012 622,312	The complete BFR Sequence =971+1232+12+(1163*94)+3, reset+1st_row+rows+I/O =1411+1760+12+(1691*138)+3 =1851+2288+12+(2219*182)+3 =2291+2800+12+(2731*226)+3

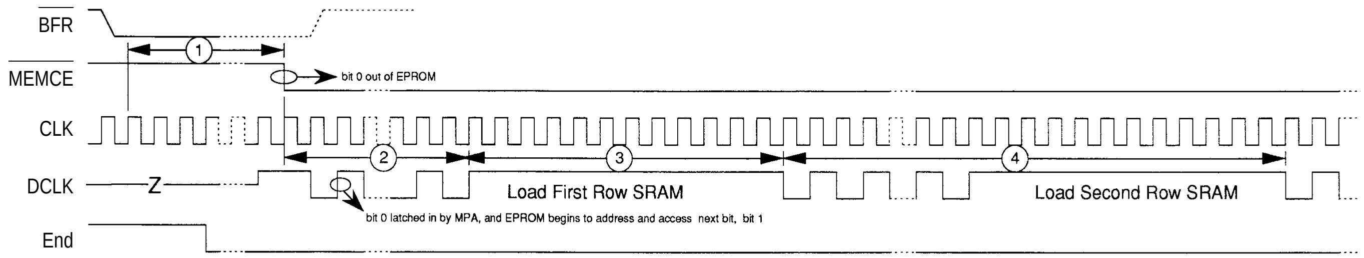


Figure 2–23. Start of a Typical Serial Boot From ROM Sequence
(Clock may be internal or external. BFR is an external asynchronous signal, T_{SU_BFR} is assumed to have been met.)

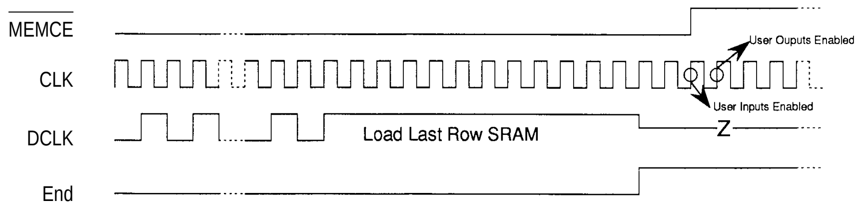


Figure 2–24. Completion of a Serial Boot From ROM Sequence

CLK cycles and transfers this data to the interal SRAM row. Provided no device ID or check–sum errors are detected, the load will continue in this row by row fashion until complete. As the last row of SRAM is written, the END signal asserts then user I/O is enabled as shown.

BFR Mode 2 Operation: 1 bit (serial) data, External Address Generation

BFR mode 2 is used for connecting MPA devices to a serial configuration memory. The MPA device provides an address increment signal (DCLK) rather than an internally generated address as in BFR mode 1. Low pin count serial memories, like the MPA17128, contain address generation logic which responds to a single increment signal. Addressing is sequential starting at zero. Multiple MPA17000 devices can be daisy chained if a larger memory is required (See MPA17128 data sheet on page NO TAG). Serial memories are programmed (written) in the opposite bit order from the way they are read. The MPA Design System configuration generation program will generate a correctly formatted PROM programming file by reflecting each configuration byte prior to writing the file.

MEMCE is high until configuration commences. MEMCE is connected to the RST/OE pin of the MPA17128 holding its internal address counter at 0 and its outputs tristated. The falling edge of MEMCE enables the memory data pin and 2 clocks later a data bit is latched into the MPA1000 device. The first rising edge of DCLK signals the memory to index its address register and present the next locations data bit. Each time 8 bits are accumulated by the MPA1000, they are written to the internal row data register. As in BFR 1, this process proceeds until a complete row is loaded and the ECB is verified. DCLK holds while the row data register is written to the current configuration memory row. After the write completes, additional bits are loaded until the next row boundary is reached. Configuration completion and error indications are identical to BFR 1.

BFR Mode 3 Operation: 8 bit data, External Address Generation

BFR mode 3 is identical to BFR mode 2 except that 8 bits of data are loaded rather than one. An external address generator is used and responds to the MPA address increment signal (DCLK). BFR mode 3 is useful because BFR 1 requires 18 user I/O signals (ADD[17:0]) during configuration. While these are subsequently released, it does impose restrictions on surrounding circuitry complicating overall system design. Secondly in applications requiring rapid configuration of a large number of MPA devices or many alternate configurations, the MPA 18 bit address space may not be large enough and an external counter (address generator) would required anyway.

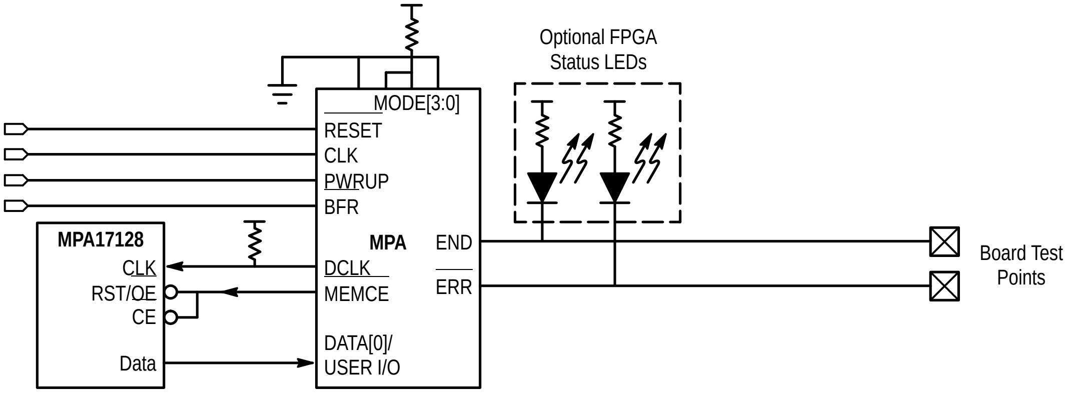


Figure 2–25. BFR Mode 2: 1–Bit (Serial) Data, External Address, External Clock

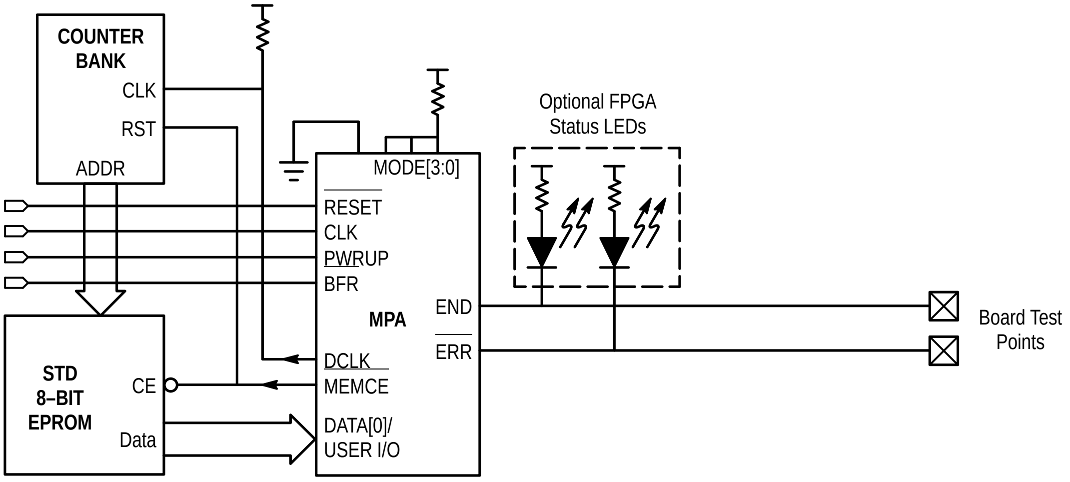
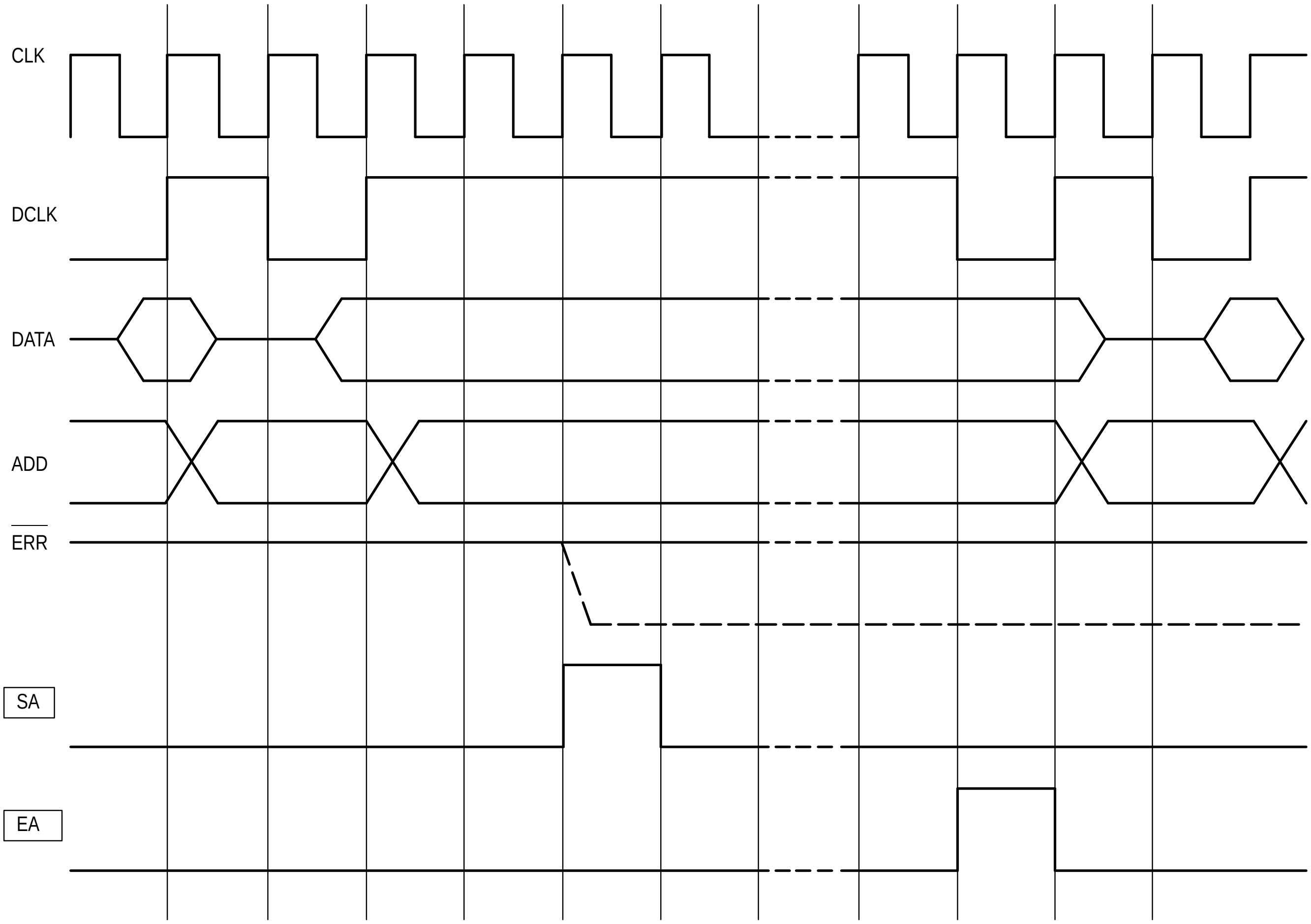


Figure 2–26. BFR Mode 3: 8–Bit Data, External Address, External Clock



SA = Start Row Access
EA = End Row Access

Figure 2–27. End of Row Behavior

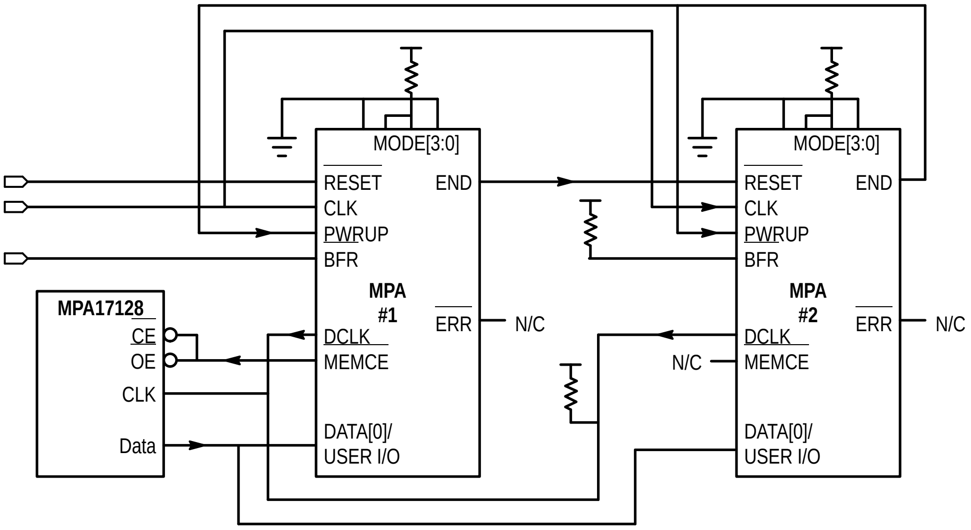


Figure 2–28. Multiple Device Subsystem: BFR2; Serial Data, External Address, External Clock

BFR Multiple Device Subsystems

If multiple devices are used together in BFR mode, the first device loads first and the END signal on each device is connected to the RESET pin of the next device. As an upstream device completes configuration, a configuration sequence is initiated on the next device. This daisy chain extends to the last device. This devices END is connected to the PWRUP pins of all subsystem MPA devices. All devices enter user mode when the last device successfully configures.

Care must be taken to insure proper operation. BFR on all but the first device must be tied high and the subsystems composite DCLK line must be pulled up to eliminate spurious clock signals as one device tristates DCLK and the next device asserts it. Figure 2–29 illustrates the control signal hand off.

When constructing a subsystem in which the first device asserts the 18 bit address (BFR mode 1), this device provides address generation for all devices in the subsystem. The DCLK pin of the first device becomes an input when it successfully configures and its internal counter remains active. Positive edges applied to this pin will increment the first devices internal address counter and present the resulting address on the first devices 18 bit address bus. Subsequent devices in this subsystem should use BFR mode 3 (external address, 8 bit data).

Examples of multiple device boot configurations are shown in Figure 2–28, Figure 2–30 and Figure 2–31. The last device's END signal is fed back into the first device's PWRUP pin. Holding PWRUP low, holds the MEMCE output low.

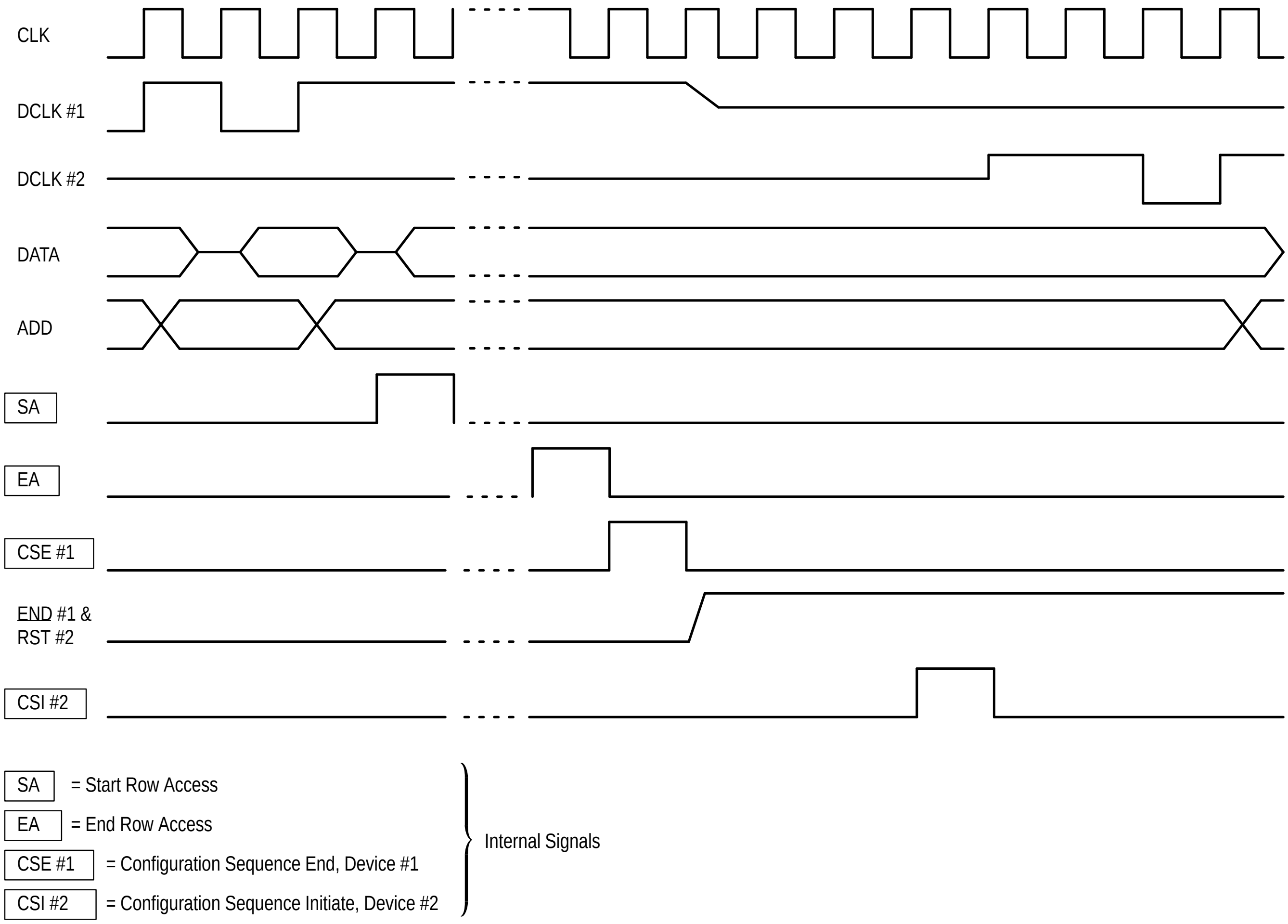
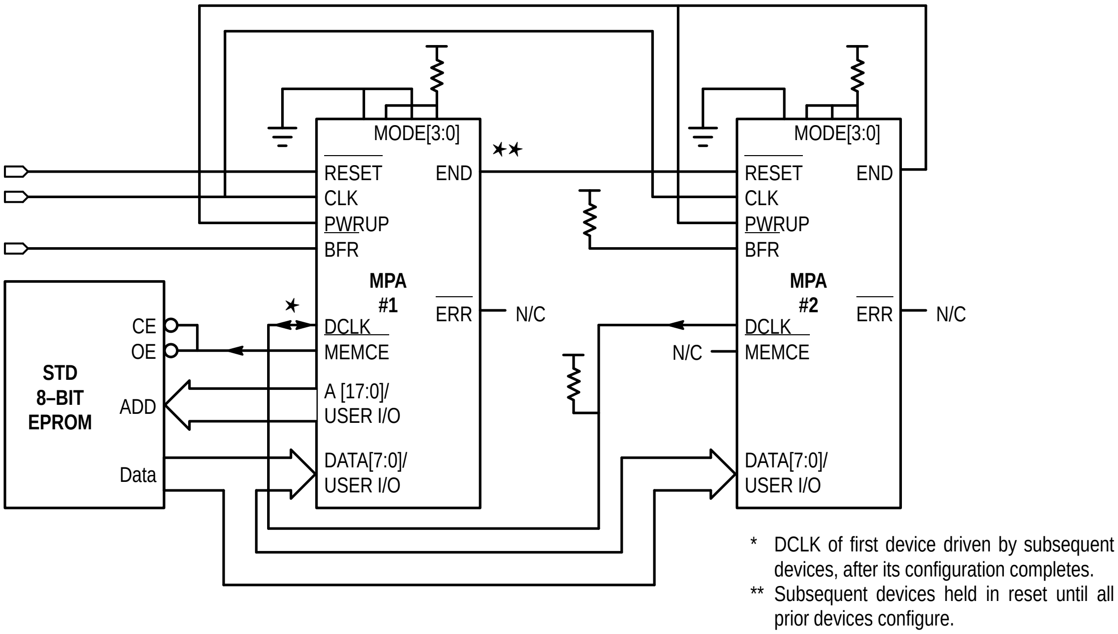


Figure 2–29. BFR Mode Daisy Chain Timing



* DCLK of first device driven by subsequent devices, after its configuration completes.

** Subsequent devices held in reset until all prior devices configure.

Figure 2–30. Multiple Device Subsystem: BFR1 and BFR3; 8 Bit Data, Internal Address, External Clock

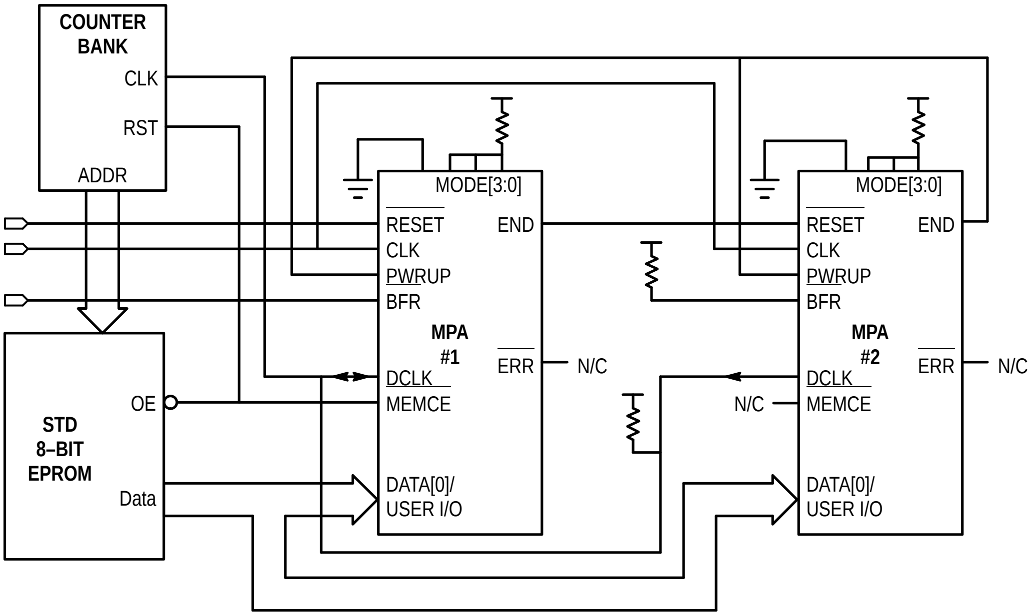


Figure 2–31. Multiple Device Subsystem: BFR3, 2; 8 Bit Data, External Address, External Clock

MICRO Mode

In MICRO Mode the MPA device behaves as an asynchronous microprocessor peripheral. Table 2–7 details MICRO Mode configuration pin function. A chip select (/CS) is derived from the processor address and enables a single MPA device. In a multiple device subsystem, a chip select for each MPA device is required. When a device is selected, the data bus is used to write commands, read status, write configuration data and read configuration data. There are two device configuration registers, the function register (RS=0) and the data/status register (RS=1). Configuration commands are written to the function register. Subsequent behavior is specific to the command issued and is documented in Table 2–8. The data register is either used to read device status, read device configuration data or write device configuration data. RS is normally connected to the least significant address line to map the function register to address A and the data/status register to address A+1.

Configuration data format information can be found in the “Device Configuration Data Format” section on page 2–32. Configuration data is generated by the MPA design system configuration generator after a layout is complete.

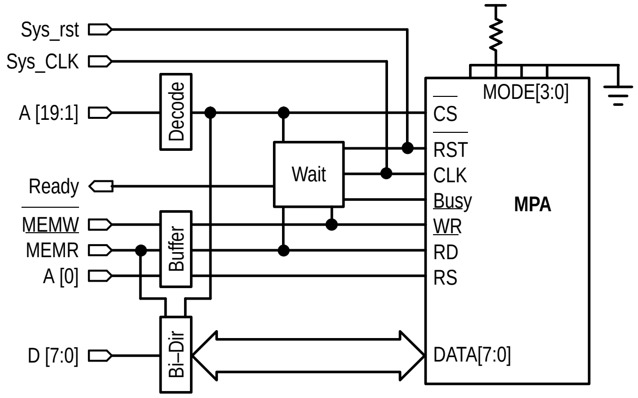


Figure 2–32. MICRO Mode: Single Device With External Clock and Wait State Insertion

Table 2–7. MICRO Mode Configuration Control Pins

Pin Name	Micro	I/O		Description
MODE[3:0]	MODE[3:0]	I	Dedicated*	**Mode Pins**
$\overline{\text{RESET}}$	$\overline{\text{RESET}}$	I	Dedicated	**General configuration reset**
CLK	CLK	I/O	Dedicated	**Clock for configuration circuitry** — If external clock is selected, pin is an input. If not selected internal configuration clock is used and output through this pin.
F0	$\overline{\text{CS}}$	I	Dedicated	**Chip select** for device in MICRO Mode.
F1	$\overline{\text{RD}}$	I	Dedicated	**Micro read signal**
F2	$\overline{\text{WR}}$	I	Dedicated	**Micro write signal**
F3	RS	I	Dedicated	**Register select** — Two register locations are active: Function Register (RS = 0) and Data/ Status Register (RS = 1).
F4	Busy	O	Dedicated	**Busy signal** — Active high when device is not ready to accept data, i.e. while device is resetting data in array or a data register to array transfer is taking place.
DATA[7:0]	DATA[7:0]	I/O	Dedicated	**Micro data port** — for configuration logic.
JTAG [4:0]	J [4:0]	I/O	User/JTAG	**JTAG pins** — JTAG or User I/O is selected by MODE[3].

* Dedicated — Pins used for configuration. Not available for user I/O.

Table 2–8. MICRO Mode Function Register (RS=0)

DATA					Function
7	**6**	**5**	**4**	**[3:0]**	
				0000	**Normal operation** — No function performed.
				0001	**Reset Device** — Entire device configuration is reset. BUSY is asserted until reset completes.
				0010	**Load Configuration** — After writing this command, an entire normal format device configuration is presented to the data register in 8 bit segments starting with the configuration header block. At any time during the loading process, a read from the data register will return status register contents. As complete rows including ECB are loaded, BUSY is temporarily asserted while row data is internally transferred from the internal data register to the currently addressed memory row. Once this write operation is complete, BUSY is deasserted and additional data can be written. Each time BUSY is deasserted, the status register should be checked for incorrect ID or row configuration data error(s). Once an error is detected, NO further write accesses to the data register will be accepted until the device is reset or another load configuration command is issued.
				0011	**Reset Row** — Indicates that the next data written to the data register will be a device row address. After the address is written, the contents of that configuration memory row are reset. BUSY is asserted after the address is written and deasserted when the operation is complete.
				0100	**Load Row** — The next data written to the data register consists of a row address followed by configuration data for that row including the terminating ECB. After the ECB is written, BUSY will be asserted during internal write and deasserted when the write completes. Reading the data register returns status register contents. The status register should be checked for row configuration data error(s). Once an error has been detected, NO further write accesses to the data register will be accepted until the device is reset or a load configuration command is issued.
				0101	**Read Row** — The next data written to the data register will be interpreted as a row address. After the row address is written, BUSY is asserted while row data is read into the internal data register. BUSY is deasserted when the transfer is completed. Subsequent successive reads from the data register will return row configuration data. No ECB is returned. The row data read back is in the same order as it is written, rightmost byte first.
				0110	**Read Device ID** — 4 subsequent reads from the data register return device ID. The most significant ID byte is read first. Refer to “configuration data format” for individual device ID values.
				0111	**Bits [3:0]** — Reserved pattern.
				1XXX	**Bits [3:0]** — Reserved pattern.
			1		**User Outputs Enabled** — Normally user outputs are enabled one or more clocks after user inputs are enabled to insure valid input values have propagated into the device.
		1			**User Inputs Enabled** — Normally user inputs are enabled after a configuration is successfully loaded into the device.
	1				**Internal Oscillator Disabled** — Normally always enabled. May be disabled if external clock and V_{PP} are user supplied. If internal configuration clock is used (Mode[2] = 0), oscillator cannot be disabled. Internal oscillator drives a charge pump that generates bootstrap V_{PP}. If turned off, then back on, allow 100µs restart time.
1					**Bootstrap Enabled (V_{PP})** — Should be enabled after configuration is completed and disabled during configuration. (Disable ties V_{PP} to V_{DD}.) Only a few package types bond out V_{PP} to a pin. The V_{PP} pin can be used to monitor V_{PP}. The V_{PP} pin may be driven between V_{DD} and 6.5V externally if Bootstrap is enabled **and** internal oscillator is disabled. V_{PP} is applied to the pass gate transistors inside the array, to ensure the lowest possible R_{DON}.

Table 2–9. MICRO Mode Data/Status Register (RS=1)

Bit Position								Function
[7]	**[6]**	**[5]**	**[4]**	**[3]**	**[2]**	**[1]**	**[0]**	
R	R	R	R	R			1	Incorrect Device ID.
R	R	R	R	R		1		Row configuration data error. ECB mismatch.
R	R	R	R	R	1			Busy signal asserted. Allows software handshaking if hardware wait states are not to used.

R = Uspecified, reserved for factory use.

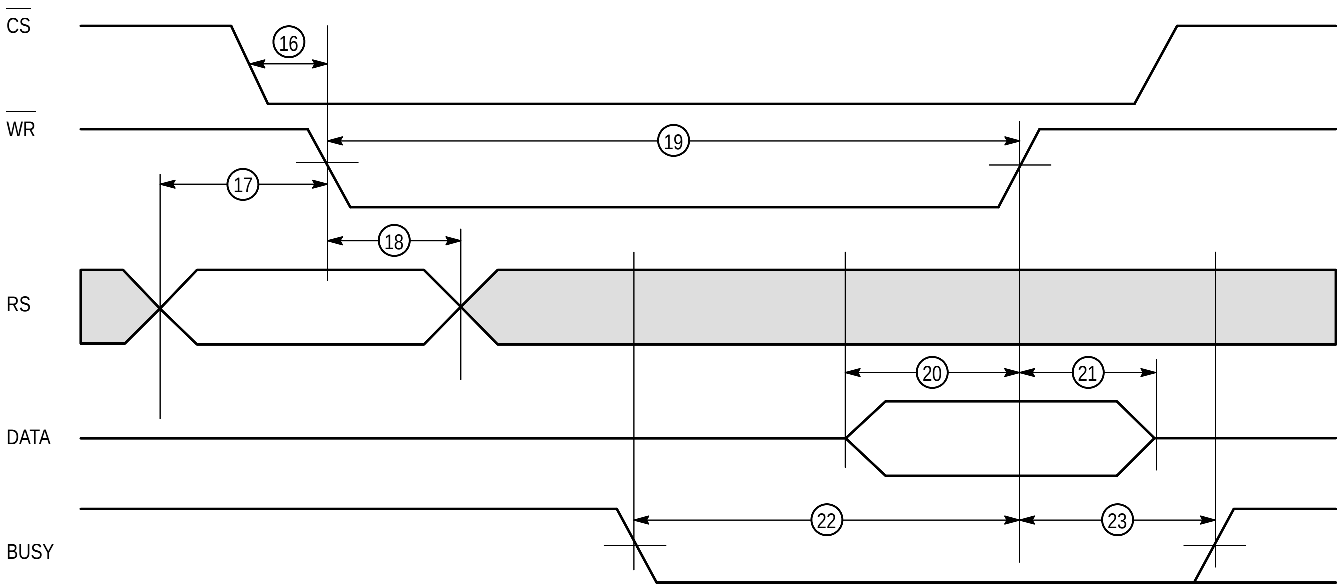


Figure 2–33. MICRO Mode External Timings (Write Cycle)

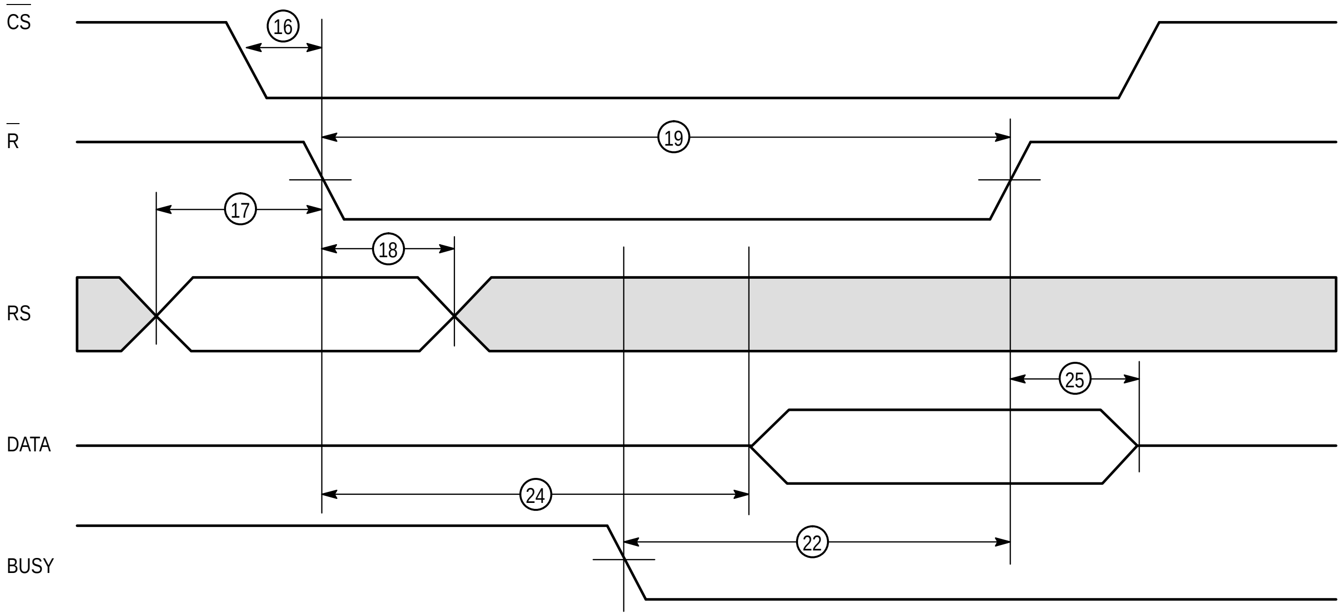


Figure 2–34. MICRO Mode External Timings (Read Cycle)

Number	Characteristic	Min	Max	Unit	Notes
16	$\overline{\text{CS}}$ Setup before Read/Write Falling Edge	10		ns	
17	RS Setup before Read/Write Falling Edge	10		ns	
18	RS Hold after Read/Write Falling Edge	10		ns	
19	Read/Write Pulse Width	50		ns	
20	Data Setup to End of Write	20		ns	
21	Data Hold after Write	10		ns	
22	Busy Inactive before End of Read/Write	50		ns	
23	Busy Active after Write	0	20	ns	
24	Data Access Time	20	40	ns	
25	Data Hold Time after Read	0	10	ns	

The configuration clock is still used to drive the MPA's internal configuration logic in MICRO mode. The length of BUSY is, therefore, a function of the configuration clock.

MICRO Mode Maximum Data Transfer Rate

The maximum MICRO Mode data transfer rate is governed by the R/W timing described in Figure 2–33 and Figure 2–34. The processor must only write data when BUSY is inactive. BUSY is only asserted when data cannot be accepted at the maximum rate. The specific behavior of BUSY for each MICRO Mode function is described in Table 2–8. When the device is powered up, an internal reset sequence is initiated and BUSY is asserted (see “Behavior During Power–On–Reset”). BUSY will be deasserted when the internal reset sequence completes. The processor can monitor BUSY directly or the status register can be read.

If processor R/W cycles are faster than the timing shown, external circuitry must be used to insert wait states. Figure 2–32 and Figure 2–39 show an application circuit consisting of one or more MPA devices and an optional wait state insertion block used to lengthen R/W timing based on CS, MEMW, MEMR, BUSY, and RESET using an externally provided clock.

Using MICRO Mode to Read Configuration SRAM

An interesting side benefit of using MICRO mode is the ability to go back to the MPA after the normal boot process completes and read back out the configuration SRAM. While under spec operating conditions, the is no possibility of configuration SRAM corruption. Some applications, however, may have out–of–spec operating conditions, such as extreme noise on power rails or rails subject to power dips. In such applications, system level health monitoring can be augmented using this SRAM read out feature. Normal MPA device operation is not disturbed during configuration SRAM reads.

Multiple Devices in MICRO Mode

If multiple devices are used in MICRO Mode, external logic is required to individually address each MPA device using CS (chip select) signals. After configuration, the processor must write bootstrap enable, enable inputs and enable outputs commands to each device. A subsystem BUSY signal can be derived by OR–ing the BUSY signals from each individual device. Refer to Figure 2–39.

Internal Clock Specification

The internal ring oscillator is a clock source with possible frequencies ranging from 10MHz to 40Mhz. This variation is expected and does not present a problem for proper charge pump or configuration operation. The internal configuration clock is derived by dividing the oscillator frequency by 8. The internal configuration clock can be used for user mode operation and is presented on the CLK pin when MODE[2] is low.

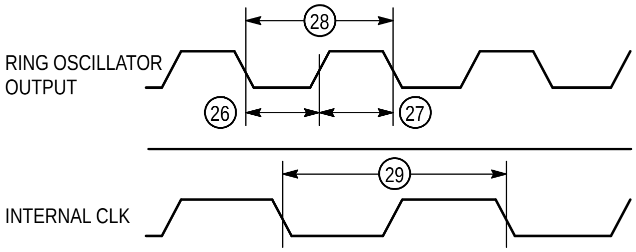
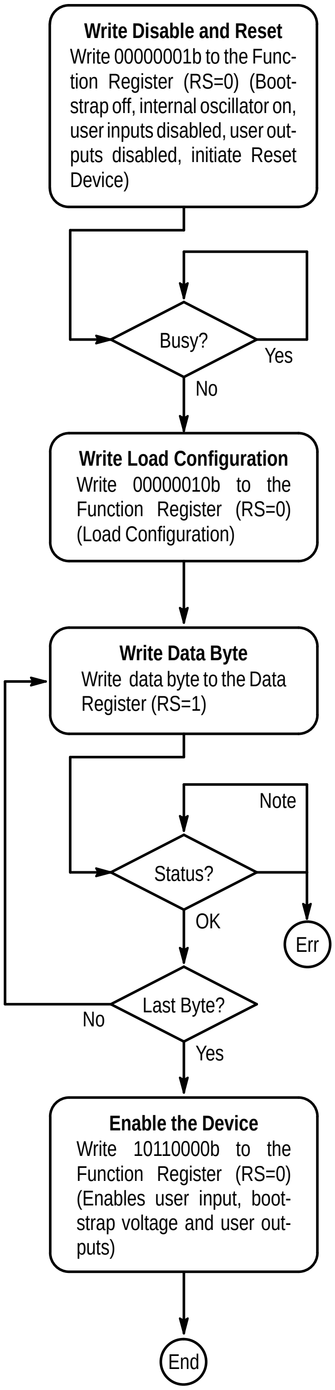


Figure 2–35. Internal Oscillator and Clock Specification

Num	Characteristic	Min	Typ	Max	Unit
26	Ring Oscillator Low	10	25	50	ns
27	Ring Oscillator High	10	25	50	ns
28	Ring Oscillator Period	25	50	100	ns
29	Internal Config Clock Period	200	400	800	ns



Busy? – There are two different ways Busy can be checked. The first is to examine the state of the physical BUSY signal. The second is to read the contents of the Data/Status Register (RS=1)

Note – The designer has a fair number of options with regards to what code to put in this "Status" test block. The code may simply check that neither busy nor any error flags have been set and move on (this is what is shown) or the code may be optimized for higher speed.

To decrease the total load time, you might want to only check for busy at the end of the every row of data. (When counting bytes to keep track of where you are in the load, remember that the first row has an additional 5 data bytes, the first four are JTAG ID and the next is the Data Type tag.)

If for some reason you also shut off the internal oscillator during this boot process, this would be the time to turn it back on (preferably prior to enabling the bootstrap; the oscillator drives the charge pump that provides bootstrap voltage). Start up time for the oscillator is not greater than 100µs.

Figure 2–36. MICRO Mode Configuration Load Sequence Example

External Clock Specification

To improve configuration performance an external clock can be connected to the CLK pin when MODE[2] is asserted. The specifications for this clock are given in Figure 2–37.

The maximum external clock frequency is 40MHz. At this frequency, boot time in the BFR modes will decrease by a factor between 8 and 32 times.

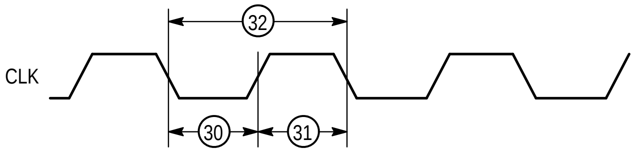


Figure 2–37. External Clock Specification

Num	Characteristic	Min	Max	Unit
30	External Clock Low	10		ns
31	External Clock High	10		ns
32	External Clock Period	25		ns

Power On Reset Operation

The MPA1000 devices contain circuitry to insure reliable self configuration when power is applied to the device. An counter clocked by the internal configuration clock and triggered by an analog power on reset circuit delays configuration until the power supply has been given sufficient settling time (Figure 2–40).

The analog power on reset circuit provides a reliable signal (APOR) to indicate that V_{DD} is sufficient to reliably operate device logic. While APOR is low a 17 bit counter is held reset. When APOR is asserted, the counter is enabled and POR occurs when the most significant counter bit reaches 1. Between APOR assertion and POR, the configuration circuitry is continuously resetting configuration memory row by row. When POR is asserted, a final internal reset sequence is performed (Figure 2–40). If an external clock is selected (by asserting MODE[2]) this final internal reset sequence will begin four internal clocks after POR, and will run using the external clock.

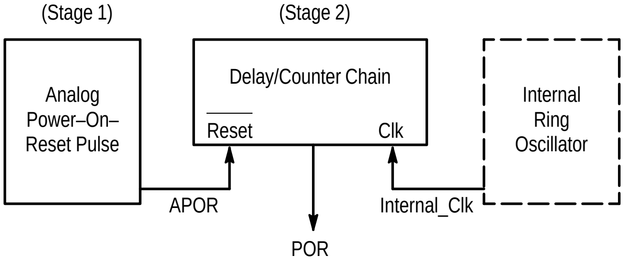


Figure 2–38. 2–Stage Power–On–Reset

External Reset

An external reset sequence can only be initiated by a falling edge on $\overline{\text{RESET}}$. If an external clock is selected (by asserting MODE[2]), it must be active in order for the reset sequence to complete successfully. Once a reset sequence is initiated it cannot be terminated by a subsequent rising edge of $\overline{\text{RESET}}$.

If $\overline{\text{RESET}}$ is low when the internal reset sequence completes, configuration will not commence until $\overline{\text{RESET}}$ is deasserted (Figure 2–41). This feature can be used to hold off configuration until other external events occur. This feature is used in conjunction with the multiple device daisy chain. Figure 2–42 shows RESET effect on other configuration mode, internal reset and internal configuration signals.

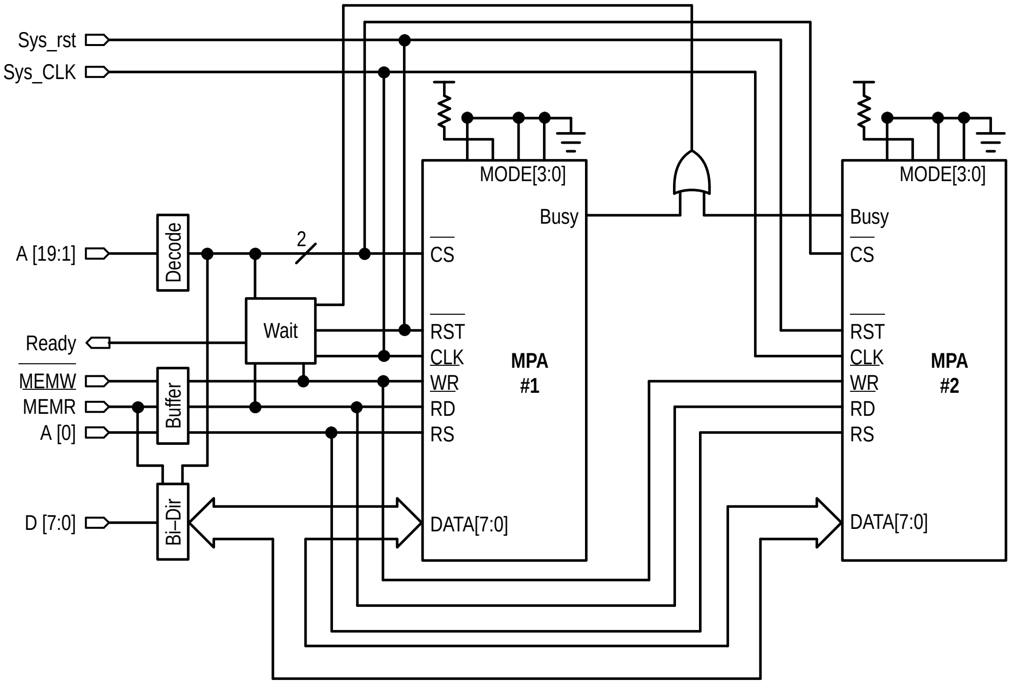


Figure 2–39. MICRO Mode: Multiple Devices With External Clock and Wait State Insertion

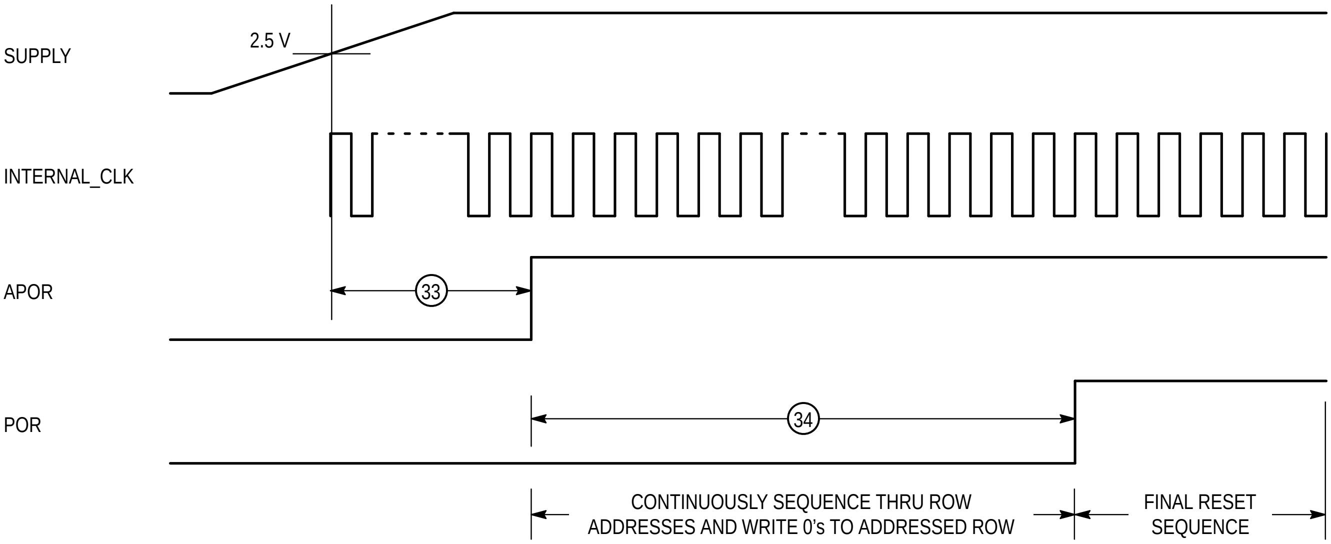


Figure 2–40. Power–On–Reset Circuitry Timing

Number	Characteristic	Min	Max	Unit	Notes
33	APOR	10	1000	µs	
34	POR (Active)	13.2	52.4	ms	

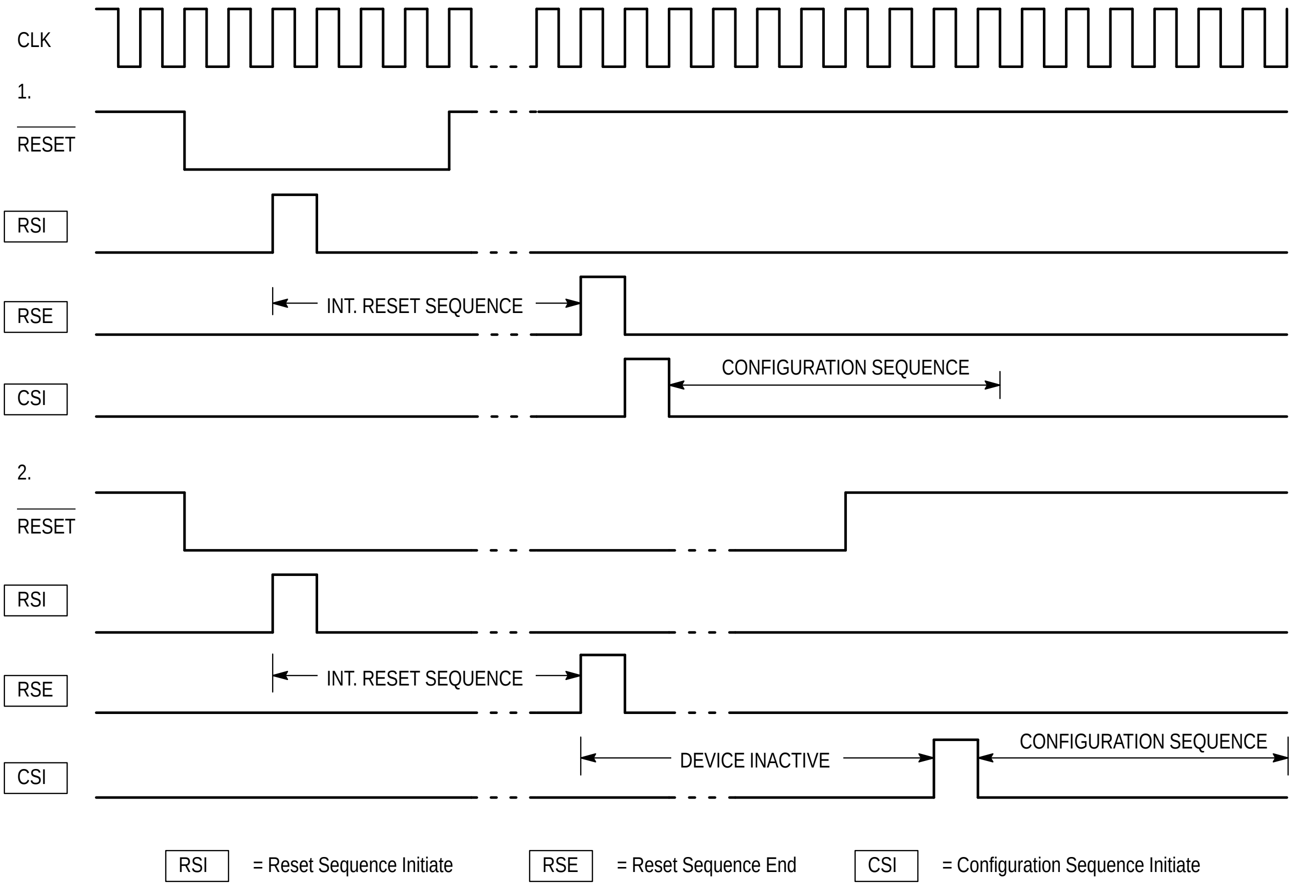
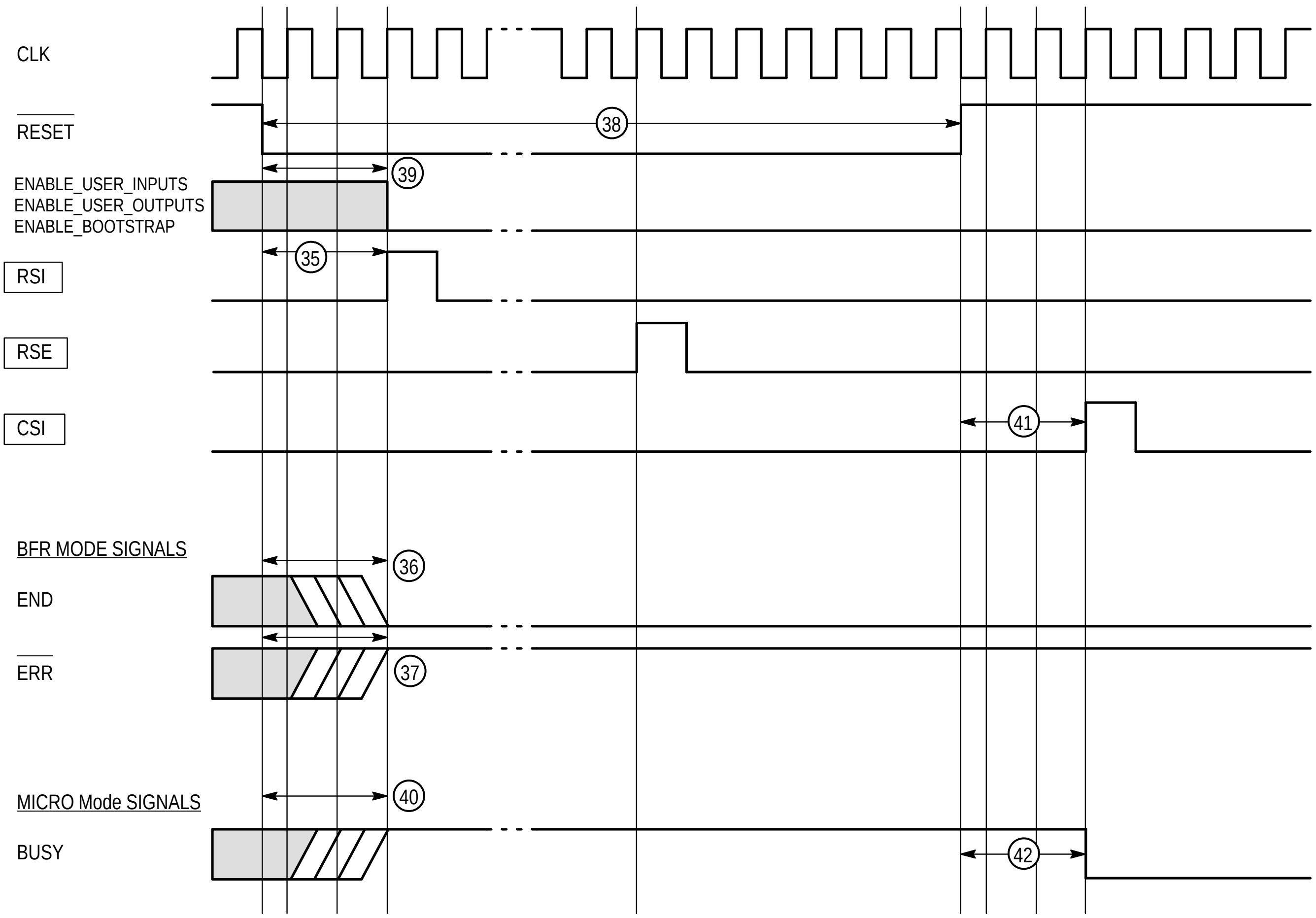


Figure 2–41. External Reset Behavior



RSI = Reset Sequence Initiate

RSE = Reset Sequence End

CSI = Configuration Sequence Initiate

Figure 2–42. External Reset Timing

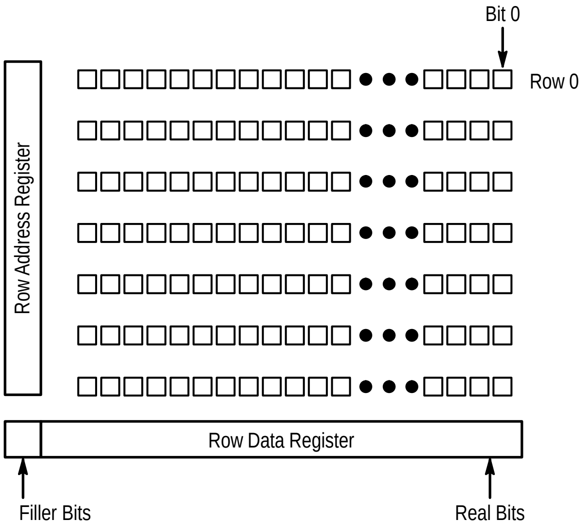
Number	Characteristic	Min	Max	Unit	Notes
35	$\overline{\text{RESET}}$ Low to Reset Sequence	2	3	CLK	
36	$\overline{\text{RESET}}$ Low to END Low	0	3	CLK	
37	$\overline{\text{RESET}}$ Low to $\overline{\text{ERR}}$ High	0	3	CLK	
38	$\overline{\text{RESET}}$ Pulse Width	50		ns	
39	$\overline{\text{RESET}}$ Low to Internal Disable	0	3	CLK	
40	$\overline{\text{RESET}}$ Low to Busy Active	0	3	CLK	
41	$\overline{\text{RESET}}$ High to CSI Pulse	2		CLK	$\overline{\text{RESET}}$ Released After RSE
42	$\overline{\text{RESET}}$ High to Busy Inactive	2		CLK	$\overline{\text{RESET}}$ Released After RSE

In MICRO Mode, the busy signal remains high while the reset signal is asserted and until the internal reset sequence is completed.

MPA1000 Configuration Data Format

Device Configuration Memory Organization

The MPA1000 devices are programmed by loading configuration data into on chip configuration memory constructed of SRAM cells. This memory is organized differently from standard memory products. The configuration SRAM is distributed throughout the MPA device. Data is read and written to the device 1 row at a time via the internal row data register (RDR). Individual rows are addressed via the row address register (RAR). Each device has a different size RDR and RAR (Figure 2–43). The configuration logic is responsible for the control of these resources.



Device	# Rows	# Real Bits	# Filler	# ECB	Total Row Bits
1016	95	562	6	8	576
1036	139	828	4	8	840
1064	183	1090	6	8	1104
1100	227	1352	0	8	1360

Figure 2–43. Device Memory Organization

A complete configuration image includes the total row bits shown in Figure 2–43, prefaced by the 5 byte header block. The total configuration imange size is given in Table 2–10.

Table 2–10. Configuration Image Size

Device	Total Bits	Decimal Bytes	Hex Bytes
1016	54,760	6,845	1ABD
1036	116,800	14,600	3908
1064	202,072	25,259	62AB
1100	308,760	38,595	96C3

Configuration logic writes data to the leftmost (most significant) RDR byte and reads from the rightmost (least significant) RDR byte. Each of these transfers occurs in 8 bit increments. When serial data is presented, the bits are accumulated into a byte before RDR transfer. Each configuration logic RDR write operation first shifts the RDR eight 8 bits and transfers the new byte into the leftmost RDR byte position. Configuration read operations transfer the rightmost RDR byte to the configuration logic and then shifts RDR contents right 8 bits.

The RAR enables a single configuration memory row. MPA configuration logic writes a row addresses into the RAR. Subsequent read or write operations are performed between the RDR and the RAR selected row in parallel.

Filler bits are used to round the RDR up to the nearest byte boundary. The ECB is not part of the RDR. During configuration a single row data vector is written to the RDR and an ECB is calculated from the data written. The calculated value is compared to the ECB contained in the data vector. If a mismatch is detected, ERR is asserted and the configuration process terminates. The ECB mechanism prevents data write disturbances from causing unpredictable device function.

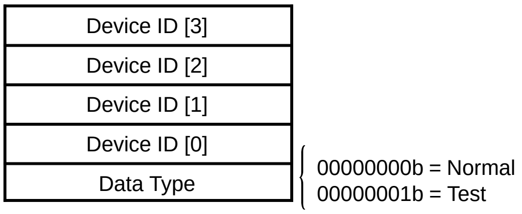
Device Configuration Data Formats

When whole configurations are loaded into a device, the first 40 bits contain a 32 bit device ID followed by an 8 bit data type field. The device ID is the same as the JTAG device ID described in “JTAG Boundary Scan”. If an incorrect ID is presented, ERR is asserted and configuration stops. Device ID comparison prevents incompatible configurations from causing unpredictable device behavior. The data type field identifies subsequent data format. Recognized data types are shown in Table 2–11.

Table 2–11.

[7:3]	[2]	[1]	[0]	Data Type
00000			0	Sequential data (Normal data)
00000			1	Test data – Multiple row access
00000		0		Unencrypted data
00000		1		Encrypted data – Not supported on first product. **Reserved** for future implementations
00000	0			Uncompressed data
00000	1			Compressed data – Not supported on first product. **Reserved** for future implementations

Header Block



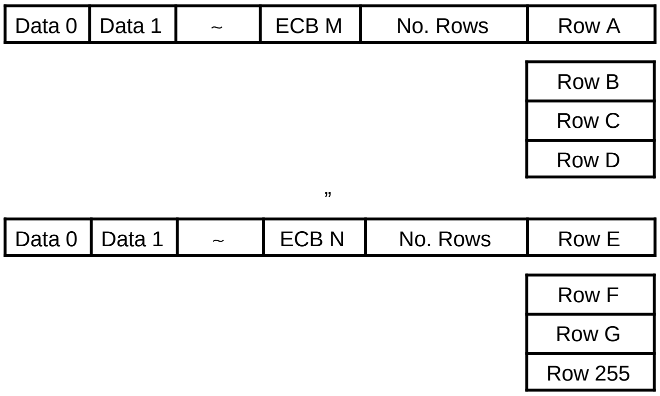
Two data formats are supported; Normal data and test data. Normal data is generated by the MPA Design System and is the only data type users are expected to use. Test data is a special format developed to aid device testing where many very regular configuration patterns must be rapidly loaded during production test. Test mode data only results in a memory savings when many rows of configuration memory contain identical information. Since this is unlikely for real designs, test mode data offers little or no benefit for reducing user configuration memory storage requirements.

Normal data consists of a series of configuration memory row images including filler bits and ECB. Each device has a different number of bytes per row and a different number of rows. A generalized normal data representation is shown in Figure 2–44. Bytes are presented to the device from left to right and from top–most row (row 0) to bottom–most row. The ECB is calculated by summing the row data byte by byte, complementing the carry and using this as the carry into the next addition.

Data 0 (*Row 0*)	Data 1 (*Row 0*)	~	~	~	ECB 0
Data 0 (*Row 1*)	Data 1 (*Row 1*)	~	~	~	ECB 1
		”	”	”	
		”	”	”	
Data 0 (*Row x*)	Data 1 (*Row x*)	~	~	~	ECB x
Data 0 (*Row y*)	Data 1 (*Row y*)	~	~	~	ECB y

Figure 2–44. Configuration Data Block (Normal Data)

Test data format is similar to normal data except that a row count and address list follows the ECB. The RDR is loaded and the ECB calculated normally. Each address is written to the RAR, a write cycle initiated to transfer the RDR to the addressed configuration memory row, the expected address count is decremented and the next address is loaded until the expected address count reaches zero. The next byte is assumed to be the first byte of a new row data vector. Configuration ends when a row address of 255 is presented. Figure 2–45 shows the generalized test data format.



(Row 255 = Configuration Terminating Byte)

Figure 2–45. Test Data Configuration

MPA1000 JTAG Boundary Scan

JTAG Boundary Scan Functions

JTAG is a standardized boundary scan methodology used for board level testing to detect faults in package and board connections, as well as internal circuitry. The MPA1000 JTAG boundary scan cell is designed to meet the IEEE std. 1149.1 for testability test of an integrated circuit.

IEEE 1149.1 Architecture

Figure 2–46 shows the general diagram of the IEEE 1149.1 MPA1000 JTAG system. Its design is compatible to Motorola H4C and H4C+ family of arrays. The MPA1000 JTAG design is hard wired.

A more detailed description of the MPA1000 JTAG system can be found in Motorola Application Note AN1618/D *Using JTAG Boundary Scan with the Motorola MPA1000 Family of FPGAs* in NO TAG on page NO TAG.

TAP and I/O Periphery Signals

The TAP (Test Access Port) consists of five externally accessible signals which are used to control and observe boundary scan data. These five pins; TCK, TMS, TDI, TRSTB, and TDO are multiplexed with normal signal pins. After JTAG testing, these pins can be programmed as normal I/O pins when MODE[3] is deasserted. The test clock pin, TCK, is used to synchronize all JTAG functions. The TCK, TMS and TRSTB control the TAP controller. TDI is the test data input pin and TDO is the test data output pin.

JTAG Control and Test Register

The **TAP Controller** is a synchronous, 16–state machine, which selects the mode of operation for the test circuitry. An example of the operation of the TAP controller is shown in Figure 2–47 where the TAP controller is sequenced through most of its test states.

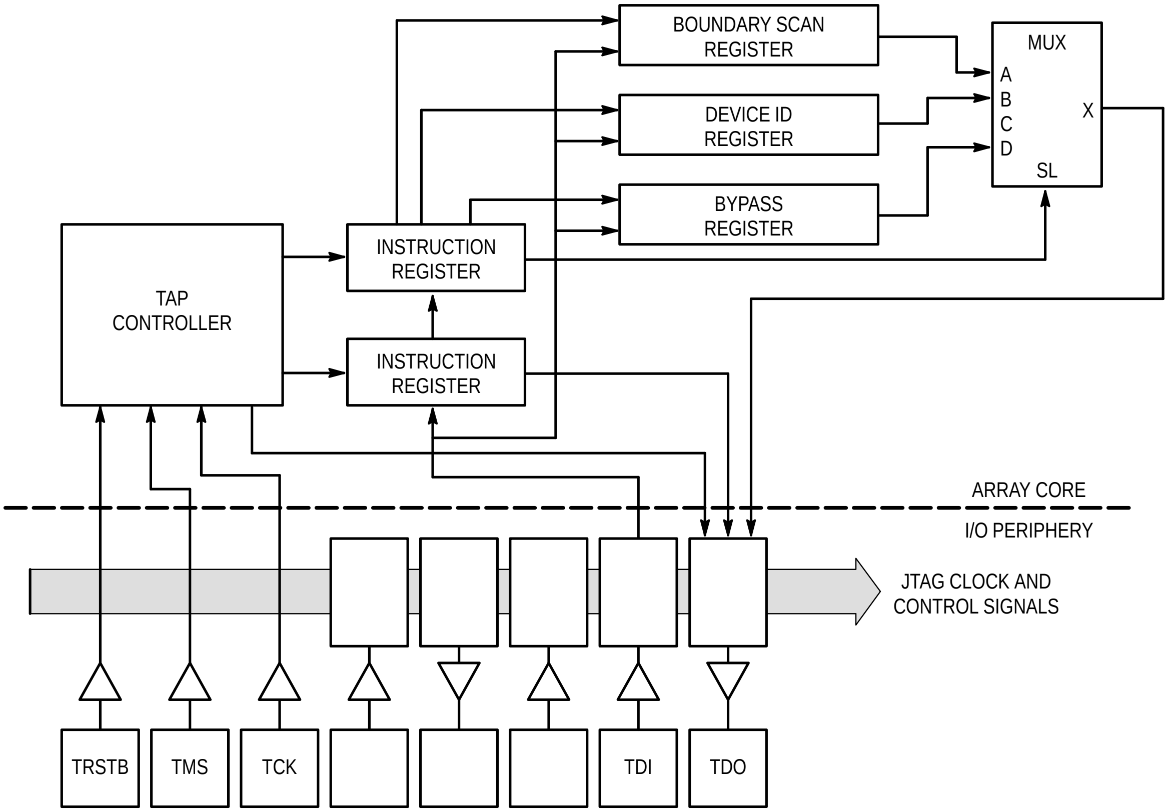


Figure 2–46. JTAG System

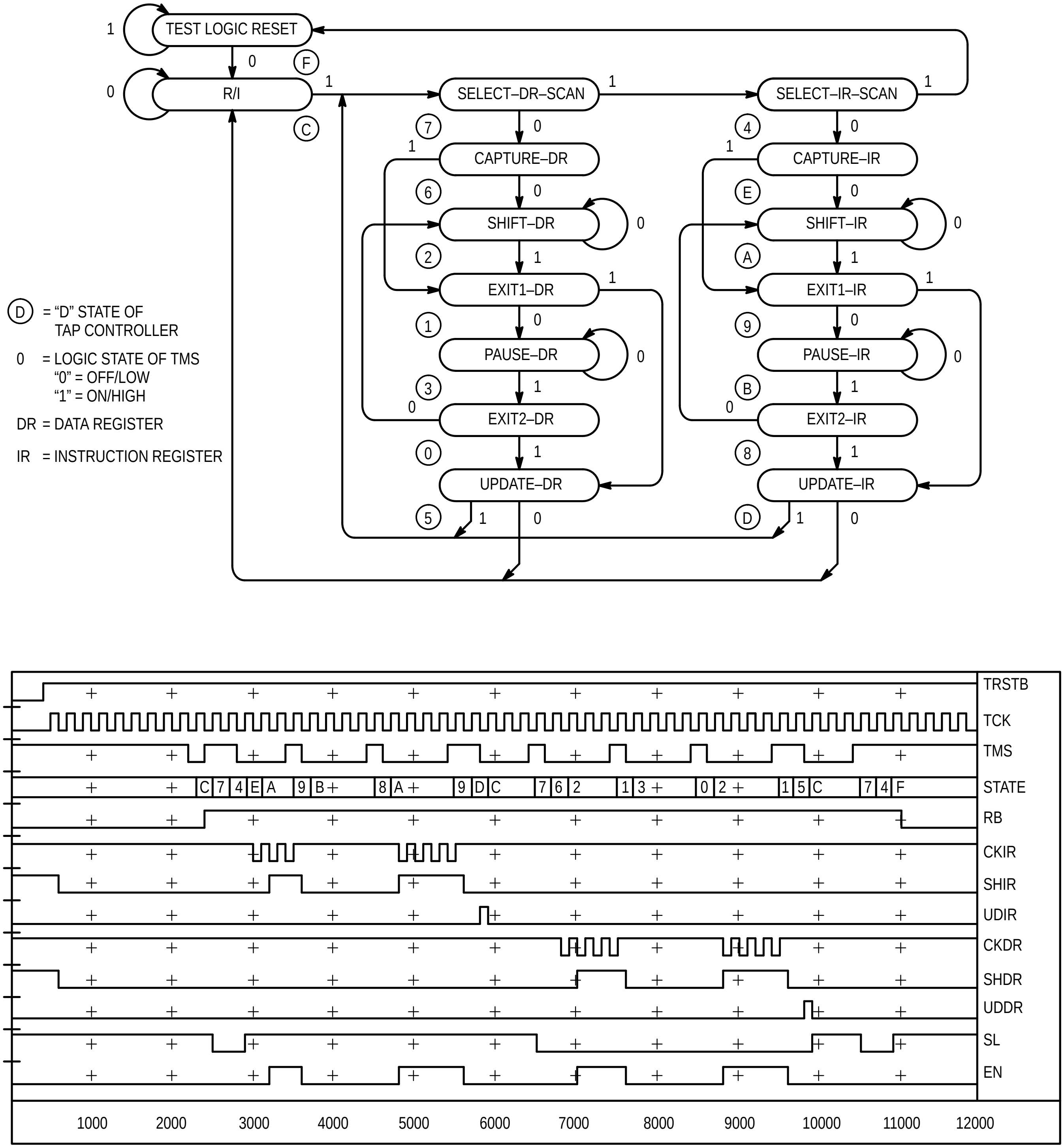


Figure 2–47. TAP Controller and Test Cycle

The **Instruction Register** is a 3–bit shift register, which permits an instruction to be shifted into the design to select the test to be performed. The **Instruction Decode** translates the instruction into separate control signals. Table 2–12 shows the basic public instructions supported by Motorola's FPGA:

Table 2–12. Basic Public Instructions

I_2	I_1	I_0	Public Instruction	Register Selected
0	0	0	EXTEST	Boundary Scan Cell
0	0	1	INTEST	Boundary Scan Cell
0	1	0	SAMPLE	Boundary Scan Cell
1	0	0	IDCODE	Device Register
1	1	1	BYPASS	Bypass Register

- **EXTEST** (external test) is the boundary scan test that checks board interconnections between integrated circuits(I.C.s).
- **INTEST** (internal test) checks the logic internal to I.C.s.
- **SAMPLE** test samples data at the I/O pins of an I.C. during normal operating mode.
- **IDCODE** instruction outputs the identification code of the I.C.
- **BYPASS** instruction redirects the test data from TDI directly to TDO, effectively removing the I.C. from the boundary scan chain.

The **Bypass Register** is a single–bit shift register used to provide a shortest path between TDI and TDO.

Table 2–13. Device Register ID Codes

Bit Number	Code Use
0–11	Motorola Identification
12–21	Array Identification
22–27	Programmable Logic Products Identification
28–31	Version Number

The **Device Identification Register** is a 32–bit register which holds a manufacturer's identity code, part number and version code. The bit assignment for the ID code is given in Table 2–13.

For example, for MPA1036 & MPA1064, the ID codes are listed as follows:

Array	**ID code** (Binary)
MPA1016	0001 001110 0100001110 000000011101
MPA1036	0001 001110 0100011110 000000011101
MPA1064	0001 001110 0100110100 000000011101
MPA1100	0001 001110 0101000000 000000011101

Array	**Hex ID code**	**Hex ID Code** (Bit Order Reversed)
MPA1016	1390E01D	C8 05 07 B8
MPA1036	1391E01D	C8 85 07 B8
MPA1064	1393401D	C8 C5 02 B8
MPA1100	1394001D	C8 25 00 B8

The JTAG ID code can easily be located when viewing a configuration file with a text editor. The ID code is always the first four data bytes. The bit order reversed version of the code shows up in configuration images targeted to serial EPROMs.

The **Boundary Scan Register** is the chain of JTAG boundary scan cells that are linked together to form a shift register around the periphery of the array. The test data enters the boundary scan register through the TDI pin, the rising edge of CKDR when SHDR is asserted, then is shifted around the array through each I/O cell in a counter clockwise direction, and finally exits through the TDO pin. Since each I/O pin is designed as a bidirectional pin, a 2–bit shift register resides in each I/O cell, one for monitor either the input or output, and the other to monitor the enable pin of the 3–state output buffer. For every two clock cycles, the data shifts from one I/O site to the other. The boundary scan cell resides in every I/O site with the exception of TDI, TCK, TMS, TRSTB and TDO pins.

MPA1000 Pin Definitions

Table 2–14. MPA1000 Package Pinout Compatibility

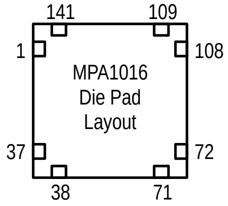
Device	FN Suffix 84–Pin PLCC	DD Suffix 128–Pin QFP	DH Suffix 160–Pin QFP	DK Suffix 208–Pin QFP
MPA1016	•	•		
MPA1036	•	•	•	
MPA1064			•	•
MPA1100				•

Table 2–15. Pin Definitions

Pin	Definition
5V Int Vdd	Internal array power (V_{DD})
5V Ext Vdd	Pad driver power for I/Os programmed to 5V
3V Ext Vdd	Pad driver power for I/Os programmed to 3V. If no I/Os are programmed to 3V, tie to 5V Ext Vdd. If 3V I/Os are used, connect is a 3.0V or 3.3V supply. These pins must be ≤ 5V Ext Vdd.
Ext Vss	Pad driver V_{SS}
Int Vss	Internal array V_{SS}
I/O	User I/O
I/O Clk	User I/O with optional clock input

MPA1000 Pin Assignments

Pinouts for MPA1016



Edge	Pad	Pad Type	Pin Location	
			FN Suffix 84–Pin PLCC	DD Suffix 128–Pin QFP
	1	5V Int Vdd		1
	2	Ext Vss		
	3	3V Ext Vdd	12	2
	4	5V Ext Vdd		
	5	Ext Vss		3
L	6	I/O (A16)	13	4
L	7	I/O (A15)	14	5
L	8	I/O (A14)	15	6
L	9	I/O (A13)	16	7
L	10	I/O (A12)	17	8
	11	5V Ext Vdd		
L	12	I/O (A11)	18	9
L	13	I/O		10
L	14	I/O (A10)	19	11
L	15	I/O		12
L	16	I/O Clk	20	13
	17	Int Vss	21	14
L	18	I/O Clk	22	15
L	19	I/O (A9)	23	16
L	20	I/O (A8)	24	17
L	21	I/O		18
L	22	I/O (A7)	25	19
	23	Ext Vss	26	20
L	24	I/O		21
L	25	I/O (A6)	27	22
L	26	I/O		23
L	27	I/O (A5)	28	24
L	28	I/O		25
	29	F[4]	29	26
	30	Int Vss		27
	31	F[3]	30	28
	32	Ext Vss		
	33	F[2]	31	29
	34	5V Ext Vdd		30
	35	F[0]	32	31
	36	3V Ext Vdd		32

Shaded areas indicate Alternate I/O Zones.

Edge	Pad	Pad Type	Pin Location	
			FN Suffix 84–Pin PLCC	DD Suffix 128–Pin QFP
	37	Ext Vss		
	38	Ext Vss		33
	39	5V Ext Vdd		
	40	$\overline{\text{RESET}}$	33	34
	41	3V Ext Vdd		35
	42	F[1]	34	36
B	43	I/O (A4)	35	37
B	44	I/O (A3)	36	38
B	45	I/O (A2)	37	39
B	46	I/O (A1)	38	40
B	47	I/O (A0)	39	41
	48	5V Ext Vdd	40	42
B	49	I/O		43
B	50	I/O		44
B	51	I/O (D7)	41	45
B	52	I/O		46
B	53	I/O Clk	42	47
	54	5V Int Vdd	43	48
B	55	I/O Clk	44	49
B	56	I/O		50
B	57	I/O		51
B	58	I/O (D6)	45	52
B	59	I/O		53
	60	Ext Vss	46	54
B	61	I/O (D5)	47	55
B	62	I/O (D4)	48	56
B	63	I/O (D3)	49	57
B	64	I/O (D2)	50	58
B	65	I/O (D1)	51	59
	66	MODE[0]	52	60
	67	Ext Vss		61
	68	MODE[1]	53	62
	69	3V Ext Vdd		
	70	5V Ext Vdd		63
	71	Probe Pad	NOT BONDED	
	72	Ext Vss		66

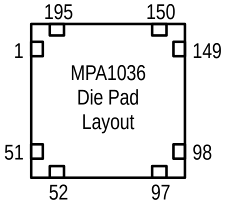
Pinouts for MPA1016 (continued)

Edge	Pad	Pad Type	Pin Location	
			FN Suffix 84–Pin PLCC	**DD Suffix 128–Pin QFP**
	73	5V Ext Vdd		
	74	MODE[2]	54	67
	75	5V Int Vdd		
	76	MODE[3]	55	68
	77	Vpp		
	78	Clk	56	69
	79	3V Ext Vdd	57	70
	80	Ext Vss		71
R	81	I/O (DCLK)	58	72
R	82	I/O		73
R	83	I/O (D0)	59	74
R	84	I/O		75
R	85	I/O (TDO)	60	76
	86	Ext Vss		77
R	87	I/O (TDI)	61	78
R	88	I/O (TMS)	62	79
R	89	I/O		80
R	90	I/O (TRSTB)	63	81
R	91	I/O Clk	64	82
	92	Int Vss	65	83
R	93	I/O Clk	66	84
R	94	I/O		85
R	95	I/O	67	86
R	96	I/O		87
R	97	I/O	68	88
	98	Ext Vss		89
R	99	I/O (TCK)	69	90
R	100	I/O	70	91
R	101	I/O	71	92
R	102	I/O	72	93
R	103	I/O	73	94
	104	5V Ext Vdd		
	105	3V Ext Vdd		
	106	Ext Vss	74	95
	107	5V Ext Vdd		
	108	5V Int Vdd		96
	109	Int Vss		97
	110	Ext Vss		

Shaded areas indicate Alternate I/O Zones.

Edge	Pad	Pad Type	Pin Location	
			FN Suffix 84–Pin PLCC	**DD Suffix 128–Pin QFP**
	111	5V Ext Vdd	75	98
	112	3V Ext Vdd		99
	113	Ext Vss		
T	114	I/O	76	100
T	115	I/O	77	101
T	116	I/O	78	102
T	117	I/O	79	103
T	118	I/O	80	104
	119	Ext Vss		105
T	120	I/O	81	106
T	121	I/O	82	107
T	122	I/O	83	108
T	123	I/O		109
T	124	I/O Clk	84	110
	125	5V Int Vdd	1	111
T	126	I/O Clk	2	112
T	127	I/O		113
T	128	I/O	3	114
T	129	I/O	4	115
T	130	I/O	5	116
	131	5V Ext Vdd		117
T	132	I/O	6	118
T	133	I/O	7	119
T	134	I/O	8	120
T	135	I/O	9	121
T	136	I/O (A17)	10	122
	137	Ext Vss		123
	138	5V Ext Vdd		124
	139	Ext Vss	11	125
	140	3V Ext Vdd		126
	141	Int Vss		127
		NC		64,65,128

Pinouts for MPA1036



Edge	Pad	Pad Type	Pin Location			
			FN Suffix 84–Pin PLCC	DD Suffix 128–Pin QFP	DH Suffix 160–Pin QFP	HI Suffix 181–Pin PGA
	1	5V Int Vdd		1		A2
	2	Ext Vss				VSSE
	3	3V Ext Vdd	12	2	40	A1
	4	5V Ext Vdd				VDDE
	5	Ext Vss		3		VSSE
L	6	I/O (A16)	13	4	39	B2
L	7	I/O			38	C2
L	8	I/O (A15)	14	5	37	D4
L	9	I/O			36	B1
L	10	I/O (A14)	15	6	35	C3
	11	5V Ext Vdd				VDDE
L	12	I/O (A13)	16	7	34	D3
L	13	I/O			33	C1
L	14	I/O (A12)	17	8	32	D2
L	15	I/O			31	D1
L	16	I/O (A11)	18	9	30	E3
	17	Ext Vss			29	VSSE
L	18	I/O		10	28	F3
L	19	I/O (A10)	19	11	27	E1
L	20	I/O			26	E2
L	21	I/O		12	25	F1
L	22	I/O Clk	20	13	24	G3
	23	5V Int Vdd			23	G1
	24	Int Vss	21	14	22	VSSI
L	25	I/O Clk	22	15	21	G2
L	26	I/O			20	F2
L	27	I/O			19	H1
L	28	I/O (A9)	23	16	18	H3
L	29	I/O			17	H2
	30	5V Ext Vdd			16	VDDE
L	31	I/O (A8)	24	17	15	J1
L	32	I/O		18	14	J2
L	33	I/O (A7)	25	19	13	K1
L	34	I/O			12	K2
L	35	I/O			11	L1
	36	Ext Vss	26	20	10	VSSE
L	37	I/O		21	9	M1
L	38	I/O (A6)	27	22	8	L2
L	39	I/O		23	7	N1
L	40	I/O (A5)	28	24	6	J3
L	41	I/O		25	5	P1
	42	F[4]	29	26	4	K3
	43	Int Vss		27		VSSI
	44	F[3]	30	28	3	M2
	45	Ext Vss				VSSE
	46	F[2]	31	29	2	L3
	47	5V Ext Vdd		30		VDDE
	48	F[0]	32	31	1	M3
	49	3V Ext Vdd		32		P2
	50	Ext Vss				VSSE
	51	Ext Vss				VSSE
	52	Ext Vss		33		VSSE
	53	5V Ext Vdd				VDDE
	54	$\overline{\text{RESET}}$	33	34	160	R1
	55	3V Ext Vdd		35		N2
	56	F[1]	34	36	159	R2
B	57	I/O (A4)	35	37	158	N3
B	58	I/O			157	R3
B	59	I/O (A3)	36	38	156	N4
B	60	I/O			155	R4
B	61	I/O (A2)	37	39	154	P3
	62	Ext Vss			153	VSSE
B	63	I/O			152	N5
B	64	I/O (A1)	38	40	151	R5
B	65	I/O			150	P4
B	66	I/O (A0)	39	41	149	R6
B	67	I/O			148	N6
	68	5V Ext Vdd	40	42	147	VDDE
B	69	I/O		43	146	P5
B	70	I/O		44	145	R7
B	71	I/O (D7)	41	45	144	N7
B	72	I/O		46	143	R8

Shaded areas indicate Alternate I/O Zones.

Pinouts for MPA1036 (continued)

Edge	Pad	Pad Type	Pin Location			
			FN Suffix 84–Pin PLCC	DD Suffix 128–Pin QFP	DH Suffix 160–Pin QFP	HI Suffix 181–Pin PGA
B	73	I/O Clk	42	47	142	N8
	74	Int Vss			141	VSSI
	75	5V Int Vdd	43	48	140	P6
B	76	I/O Clk	44	49	139	P8
B	77	I/O		50	138	P7
B	78	I/O		51	137	R9
B	79	I/O (D6)	45	52	136	P9
B	80	I/O		53	135	R10
	81	Ext Vss	46	54	134	VSSE
B	82	I/O (D5)	47	55	133	R11
B	83	I/O			132	N9
B	84	I/O (D4)	48	56	131	R12
B	85	I/O			130	P10
B	86	I/O (D3)	49	57	129	P11
	87	5V Ext Vdd				VDDE
B	88	I/O			128	R13
B	89	I/O (D2)	50	58	127	N10
B	90	I/O			126	R14
B	91	I/O (D1)	51	59	125	N11
B	92	I/O			124	P13
	93	MODE[0]	52	60	123	P12
	94	Ext Vss		61		VSSE
	95	MODE[1]	53	62	122	N12
	96	3V Ext Vdd			121	P14
	97	5V Ext Vdd		63		VDDE
	98	Probe Pad	NOT BONDED			
	99	Ext Vss				VSSE
	100	Ext Vss		66		VSSE
	101	5V Ext Vdd				VDDE
	102	MODE[2]	54	67	120	M12
	103	5V Int Vdd				R15
	104	MODE[3]	55	68	119	N13
	105	Vpp				P15
	106	Clk	56	69	118	L13
	107	3V Ext Vdd	57	70	117	N15
	108	Ext Vss		71		VSSE

Shaded areas indicate Alternate I/O Zones.

Edge	Pad	Pad Type	Pin Location			
			FN Suffix 84–Pin PLCC	DD Suffix 128–Pin QFP	DH Suffix 160–Pin QFP	HI Suffix 181–Pin PGA
R	109	I/O (Dclk)	58	72	116	L14
R	110	I/O		73	115	M13
R	111	I/O (D0)	59	74	114	M15
R	112	I/O		75	113	N14
R	113	I/O (Tdo)	60	76	112	K14
	114	Ext Vss		77	111	VSSE
R	115	I/O (Tdi)	61	78	110	L15
R	116	I/O			109	K13
R	117	I/O			108	K15
R	118	I/O (Tms)	62	79	107	M14
R	119	I/O			106	J15
	120	5V Ext Vdd			105	VDDE
R	121	I/O		80	104	H14
R	122	I/O (Trstb)	63	81	103	J13
R	123	I/O			102	H15
R	124	I/O			101	J14
R	125	I/O Clk	64	82	100	G14
	126	Int Vss	65	83	99	VSSI
	127	5V Int Vdd			98	G15
R	128	I/O Clk	66	84	97	H13
R	129	I/O		85	96	F15
R	130	I/O	67	86	95	G13
R	131	I/O		87	94	E15
R	132	I/O	68	88	93	F14
	133	Ext Vss		89	92	VSSE
R	134	I/O (Tck)	69	90	91	F13
R	135	I/O			90	D15
R	136	I/O	70		89	E14
R	137	I/O		91	88	C15
R	138	I/O	71		87	E13
	139	5V Ext Vdd				VDDE
R	140	I/O		92	86	D13
R	141	I/O	72		85	D14
R	142	I/O		93	84	C13
R	143	I/O			83	B15
R	144	I/O	73	94	82	D12

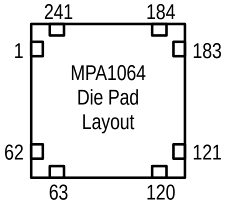
Pinouts for MPA1036 (continued)

Edge	Pad	Pad Type	Pin Location			
			FN Suffix 84–Pin PLCC	DD Suffix 128–Pin QFP	DH Suffix 160–Pin QFP	HI Suffix 181–Pin PGA
	145	Ext Vss				VSSE
	146	3V Ext Vdd				C12
	147	Ext Vss	74	95	81	VSSE
	148	5V Ext Vdd				VDDE
	149	5V Int Vdd		96		C14
	150	Int Vss		97		VSSI
	151	Ext Vss				VSSE
	152	5V Ext Vdd	75	98	80	VDDE
	153	3V Ext Vdd		99		A15
	154	Ext Vss			79	VSSE
T	155	I/O	76	100	78	B14
T	156	I/O			77	C11
T	157	I/O	77	101	76	B13
T	158	I/O			75	B12
T	159	I/O	78	102	74	A14
	160	5V Ext Vdd			73	VDDE
T	161	I/O	79	103	72	A13
T	162	I/O			71	C10
T	163	I/O	80		70	A12
T	164	I/O		104	69	B11
T	165	I/O	81		68	A11
	166	Ext Vss		105	67	VSSE
T	167	I/O	82	106	66	A10
T	168	I/O		107	65	B10
T	169	I/O	83	108	64	A9
T	170	I/O		109	63	C9
T	171	I/O Clk	84	110	62	B8
	172	5V Int Vdd	1	111	61	B9
	173	Int Vss			60	VSSI
T	174	I/O Clk	2	112	59	C8
T	175	I/O		113	58	A8
T	176	I/O	3	114	57	B7
T	177	I/O		115	56	A7
T	178	I/O	4	116	55	C7
	179	5V Ext Vdd		117		VDDE
T	180	I/O	5		54	B6
T	181	I/O		118	53	A6
T	182	I/O	6		52	C6
T	183	I/O			51	A5
T	184	I/O	7	119	50	B5
	185	Ext Vss			49	VSSE
T	186	I/O	8	120	48	C5
T	187	I/O			47	A4
T	188	I/O	9	121	46	B4
T	189	I/O			45	A3
T	190	I/O (A17)	10	122	44	C4
	191	Ext Vss		123	43	VSSE
	192	5V Ext Vdd		124	42	VDDE
	193	Ext Vss	11	125		VSSE
	194	3V Ext Vdd		126	41	B3
	195	Int Vss		127		VSSI
		NC		64,65, 128		E5

Shaded areas indicate Alternate I/O Zones.

181PGA NOTES:

VSSE Plane:	G12, E12, K12, D10, M10, G4, E4, K4, D6, M6
VSSI Plane:	E8, L8, H11, M11, H5, D5
VDDE Plane:	D8, M8, H12, F12, J12, L12, D9, M9, D11, H4, F4, M4, J4, L4, D7, M7, M5



Pinouts for MPA1064

Edge	Pad	Pad Type	Pin Location		
			DH Suffix 160–Pin QFP	DK Suffix 208–Pin QFP	KE Suffix 224–Pin PGA
	1	5V Int Vdd			VDDI
	2	Ext Vss			VSSE
	3	3V Ext Vdd	40		E4
	4	5V Ext Vdd			VDDE
	5	Ext Vss			VSSE
L	6	I/O (A16)	39	1	C4
L	7	I/O		2	B2
L	8	I/O		3	D4
L	9	I/O	38	4	C2
L	10	I/O		5	C3
	11	Ext Vss		6	VSSE
L	12	I/O (A15)	37	7	D3
L	13	I/O		8	B1
L	14	I/O	36	9	D2
L	15	I/O		10	C1
L	16	I/O (A14)	35	11	G4
	17	5V Ext Vdd		12	VDDE
L	18	I/O (A13)	34	13	E3
L	19	I/O	33	14	D1
L	20	I/O (A12)	32	15	E2
L	21	I/O	31	16	E1
L	22	I/O (A11)	30	17	F3
	23	Ext Vss	29	18	VSSE
L	24	I/O	28	19	G3
L	25	I/O (A10)	27	20	F1
L	26	I/O	26	21	G2
L	27	I/O	25	22	G1
L	28	I/O Clk	24	23	J4
	29	5V Int Vdd	23	24	VDDI
	30	Int Vss	22	25	VSSI
L	31	I/O Clk	21	26	H1
L	32	I/O		27	H3
L	33	I/O	20	28	J2
L	34	I/O		29	H2
L	35	I/O	19	30	K1
	36	Ext Vss		31	VSSE
L	37	I/O		32	L1
L	38	I/O (A9)	18	33	J3
L	39	I/O		34	L2
L	40	I/O	17	35	K3
L	41	I/O		36	M1
	42	5V Ext Vdd	16	37	VDDE
L	43	I/O (A8)	15	38	N1
L	44	I/O	14	39	K2
L	45	I/O (A7)	13	40	P1
L	46	I/O	12	41	L3
L	47	I/O	11	42	N2
	48	Ext Vss	10	43	VSSE
L	49	I/O	9	44	R1
L	50	I/O (A6)	8	45	M3
L	51	I/O	7	46	T1
L	52	I/O (A5)	6	47	L4
L	53	I/O	5	48	R2
	54	F[4]	4	49	N3
	55	Int Vss			VSSI
	56	F[3]	3	50	P2
	57	Ext Vss			VSSE
	58	F[2]	2	51	P3
	59	5V Ext Vdd			VDDE
	60	F[0]	1	52	P4
	61	3V Ext Vdd			N4
	62	Ext Vss			VSSE
	63	Ext Vss		53	VSSE
	64	5V Ext Vdd			VDDE
	65	$\overline{\text{RESET}}$	160	54	R3
	66	3V Ext Vdd		55	P5
	67	F[1]	159	56	T2
B	68	I/O (A4)	158	57	R4
B	69	I/O		58	T3
B	70	I/O	157	59	P6
B	71	I/O		60	U2
B	72	I/O (A3)	156	61	T4

Shaded areas indicate Alternate I/O Zones.

Pinouts for MPA1064 (continued)

Edge	Pad	Pad Type	Pin Location		
			DH Suffix 160–Pin QFP	DK Suffix 208–Pin QFP	KE Suffix 224–Pin PGA
	73	Ext Vss			VSSE
B	74	I/O		62	R5
B	75	I/O	155	63	U3
B	76	I/O		64	T5
B	77	I/O (A2)	154	65	U4
B	78	I/O		66	P7
	79	Ext Vss	153	67	VSSE
B	80	I/O	152	68	R6
B	81	I/O (A1)	151	69	U5
B	82	I/O	150	70	R7
B	83	I/O (A0)	149	71	U6
B	84	I/O	148	72	P8
	85	5V Ext Vdd	147	73	VDDE
B	86	I/O	146	74	T7
B	87	I/O	145	75	U7
B	88	I/O (D7)	144	76	R8
B	89	I/O	143	77	U8
B	90	I/O Clk	142	78	T8
	91	Int Vss	141	79	VSSI
	92	5V Int Vdd	140	80	VDDI
B	93	I/O Clk	139	81	T9
B	94	I/O	138	82	R9
B	95	I/O	137	83	U10
B	96	I/O (D6)	136	84	R10
B	97	I/O	135	85	T10
	98	Ext Vss	134	86	VSSE
B	99	I/O (D5)	133	87	U11
B	100	I/O	132	88	P10
B	101	I/O (D4)	131	89	T11
B	102	I/O	130	90	R11
B	103	I/O (D3)	129	91	U12
	104	5V Ext Vdd			VDDE
B	105	I/O	128	92	U13
B	106	I/O		93	P11
B	107	I/O (D2)	127	94	U14
B	108	I/O		95	R12

Edge	Pad	Pad Type	Pin Location		
			DH Suffix 160–Pin QFP	DK Suffix 208–Pin QFP	KE Suffix 224–Pin PGA
B	109	I/O	126	96	T13
	110	Ext Vss		97	VSSE
B	111	I/O		98	U15
B	112	I/O (D1)	125	99	R13
B	113	I/O		100	U16
B	114	I/O	124	101	T14
B	115	I/O		102	T15
	116	MODE[0]	123	103	R14
	117	Ext Vss			VSSE
	118	MODE[1]	122	104	R15
	119	3V Ext Vdd	121		P12
	120	5V Ext Vdd			VDDE
	121	Probe Pad	NOT BONDED		
	122	Ext Vss			VSSE
	123	5V Ext Vdd		105	VDDE
	124	MODE[2]	120	106	T16
	125	5V Int Vdd			P13
	126	MODE[3]	119	107	T17
	127	Vpp			P14
	128	Clk	118	108	P16
	129	3V Ext Vdd	117	109	N14
	130	Ext Vss		110	VSSE
R	131	I/O (DCLK)	116	111	R16
R	132	I/O		112	R17
R	133	I/O	115	113	L14
R	134	I/O		114	N16
R	135	I/O (D0)	114	115	P15
	136	Ext Vss			VSSE
R	137	I/O	113	116	N15
R	138	I/O		117	P17
R	139	I/O		118	M15
R	140	I/O		119	N17
R	141	I/O (TDO)	112	120	L15
	142	Ext Vss	111	121	VSSE
R	143	I/O (TDI)	110	122	L16
R	144	I/O	109	123	M17

Shaded areas indicate Alternate I/O Zones.

Pinouts for MPA1064 (continued)

Edge	Pad	Pad Type	Pin Location		
			DH Suffix 160–Pin QFP	DK Suffix 208–Pin QFP	KE Suffix 224–Pin PGA
R	145	I/O	108	124	K15
R	146	I/O (TMS)	107	125	L17
R	147	I/O	106	126	K16
	148	5V Ext Vdd	105	127	VDDE
R	149	I/O	104	128	J15
R	150	I/O (TRSTB)	103	129	K17
R	151	I/O	102	130	J14
R	152	I/O	101	131	J16
R	153	I/O Clk	100	132	H16
	154	Int Vss	99	133	VSSI
	155	5V Int Vdd	98	134	VDDI
R	156	I/O Clk	97	135	H17
R	157	I/O	96	136	H15
R	158	I/O	95	137	G17
R	159	I/O	94	138	G16
R	160	I/O	93	139	F17
	161	Ext Vss	92	140	VSSE
R	162	I/O		141	E17
R	163	I/O		142	G15
R	164	I/O (TCK)	91	143	D17
R	165	I/O	90	144	F15
R	166	I/O	89	145	C17
	167	5V Ext Vdd		146	VDDE
R	168	I/O	88	147	E16
R	169	I/O		148	G14
R	170	I/O	87	149	D16
R	171	I/O	86	150	E15
R	172	I/O	85	151	B17
	173	Ext Vss			VSSE
R	174	I/O	84	152	C16
R	175	I/O		153	D15
R	176	I/O	83	154	B16
R	177	I/O		155	D14
R	178	I/O	82	156	C15
	179	Ext Vss			VSSE
	180	3V Ext Vdd			E14

Edge	Pad	Pad Type	Pin Location		
			DH Suffix 160–Pin QFP	DK Suffix 208–Pin QFP	KE Suffix 224–Pin PGA
	181	Ext Vss	81		VSSE
	182	5V Ext Vdd			VDDE
	183	5V Int Vdd			VDDI
	184	Int Vss			VSSI
	185	Ext Vss		157	VSSE
	186	5V Ext Vdd	80		VDDE
	187	3V Ext Vdd		158	D13
	188	Ext Vss	79		VSSE
T	189	I/O		159	C14
T	190	I/O	78	160	B15
T	191	I/O		161	D12
T	192	I/O	77	162	A16
T	193	I/O	76	163	C13
	194	Ext Vss		164	VSSE
T	195	I/O		165	B13
T	196	I/O	75	166	B14
T	197	I/O		167	D11
T	198	I/O	74	168	A15
T	199	I/O		169	C12
	200	5V Ext Vdd	73	170	VDDE
T	201	I/O	72	171	C11
T	202	I/O	71	172	A14
T	203	I/O	70	173	B11
T	204	I/O	69	174	A13
T	205	I/O	68	175	C10
	206	Ext Vss	67	176	VSSE
T	207	I/O	66	177	B10
T	208	I/O	65	178	A12
T	209	I/O	64	179	C9
T	210	I/O	63	180	A11
T	211	I/O Clk	62	181	D9
	212	5V Int Vdd	61	182	VDDI
	213	Int Vss	60	183	VSSI
T	214	I/O Clk	59	184	A10
T	215	I/O	58	185	B9
T	216	I/O	57	186	A8

Shaded areas indicate Alternate I/O Zones.

Pinouts for MPA1064 (continued)

Edge	Pad	Pad Type	Pin Location		
			DH Suffix 160–Pin QFP	DK Suffix 208–Pin QFP	KE Suffix 224–Pin PGA
T	217	I/O	56	187	B8
T	218	I/O	55	188	A7
	219	5V Ext Vdd		189	VDDE
T	220	I/O	54	190	B7
T	221	I/O	53	191	C8
T	222	I/O	52	192	A6
T	223	I/O	51	193	C7
T	224	I/O	50	194	A5
	225	Ext Vss	49	195	VSSE
T	226	I/O		196	A4
T	227	I/O	48	197	D7
T	228	I/O		198	B5
T	229	I/O	47	199	C6
T	230	I/O	46	200	A3
	231	Ext Vss			VSSE
T	232	I/O		201	B4

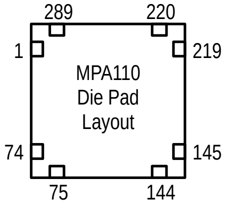
Shaded areas indicate Alternate I/O Zones.

Edge	Pad	Pad Type	Pin Location		
			DH Suffix 160–Pin QFP	DK Suffix 208–Pin QFP	KE Suffix 224–Pin PGA
T	233	I/O	45	202	D6
T	234	I/O		203	A2
T	235	I/O (A17)	44	204	C5
T	236	I/O		205	B3
	237	Ext Vss	43		VSSE
	238	5V Ext Vdd	42		VDDE
	239	Ext Vss		206	VSSE
	240	3V Ext Vdd	41	207	D5
	241	Int Vss		208	VSSI

224 PGA NOTES:

VSSI Plane:	E8, E10, H5, J13, K5, N9
VSSE Plane:	A1, A9, A17, E7, E9, E11, F4, F14, H13, J1, J5, J17, K13, M4, M14, N7, N11, P9, U1, U9, U17
VDDI Plane:	D8, D10, K4, K14, N8
VDDE Plane:	B6, B12, F2, F16, H4, H14, M2, M16, N10, T6, T12

Pinouts for MPA1100



Edge	Pad	Pad Type	Pin Location	
			DK Suffix 208–Pin QFP	HV Suffix 299–Pin PGA
	1	5V Int Vdd		VDDI
	2	Ext Vss		VSSE
	3	3V Ext Vdd		F5
	4	5V Ext Vdd		VDDE
	5	Ext Vss		VSSE
L	6	I/O		C2
L	7	I/O		D3
L	8	I/O (A16)	1	B1
L	9	I/O		E3
L	10	I/O	2	C1
	11	Ext Vss		VSSE
L	12	I/O	3	D1
L	13	I/O		F3
L	14	I/O	4	E2
L	15	I/O		G4
L	16	I/O	5	E1
	17	Ext Vss	6	VSSE
L	18	I/O (A15)	7	F2
L	19	I/O	8	H4
L	20	I/O	9	F1
L	21	I/O	10	H3
L	22	I/O (A14)	11	G2
	23	5V Ext Vdd	12	VDDE
L	24	I/O (A13)	13	G1
L	25	I/O	14	J4
L	26	I/O (A12)	15	H2
L	27	I/O	16	J3
L	28	I/O (A11)	17	H1
	29	Ext Vss	18	VSSE
L	30	I/O	19	J1
L	31	I/O (A10)	20	K4
L	32	I/O	21	K2
L	33	I/O	22	K3
L	34	I/O Clk	23	K1
	35	5V Int Vdd	24	VDDI
	36	Int Vss	25	VSSI

Edge	Pad	Pad Type	Pin Location	
			DK Suffix 208–Pin QFP	HV Suffix 299–Pin PGA
L	37	I/O Clk	26	L3
L	38	I/O	27	L1
L	39	I/O	28	L4
L	40	I/O	29	L2
L	41	I/O	30	M2
	42	Ext Vss	31	VSSE
L	43	I/O	32	M3
L	44	I/O (A9)	33	M1
L	45	I/O	34	M4
L	46	I/O	35	N1
L	47	I/O	36	N2
	48	5V Ext Vdd	37	VDDE
L	49	I/O (A8)	38	N3
L	50	I/O	39	P1
L	51	I/O (A7)	40	N4
L	52	I/O	41	P2
L	53	I/O	42	P3
	54	Ext Vss	43	VSSE
L	55	I/O	44	P4
L	56	I/O		R1
L	57	I/O (A6)	45	R3
L	58	I/O		R2
L	59	I/O	46	R4
	60	Ext Vss		VSSE
L	61	I/O		T3
L	62	I/O (A5)	47	T1
L	63	I/O		T4
L	64	I/O	48	T2
L	65	I/O		U3
	66	F[4]	49	U1
	67	Int Vss		VSSI
	68	F[3]	50	V1
	69	Ext Vss		VSSE
	70	F[2]	51	W1
	71	5V Ext Vdd		VDDE
	72	F[0]	52	V2

Shaded areas indicate Alternate I/O Zones.

Pinouts for MPA1100 (continued)

Edge	Pad	Pad Type	Pin Location	
			DK Suffix 208–Pin QFP	HV Suffix 299–Pin PGA
	73	3V Ext Vdd		R5
	74	Ext Vss		VSSE
	75	Ext Vss	53	VSSE
	76	5V Ext Vdd		VDDE
	77	$\overline{\text{RESET}}$	54	V3
	78	3V Ext Vdd	55	T6
	79	F[1]	56	U5
B	80	I/O		W2
B	81	I/O		V5
B	82	I/O (A4)	57	Y2
B	83	I/O		V6
B	84	I/O	58	Y3
	85	Ext Vss		VSSE
B	86	I/O	59	Y4
B	87	I/O	60	U7
B	88	I/O		W5
B	89	I/O (A3)	61	V7
B	90	I/O		Y5
	91	Ext Vss		VSSE
B	92	I/O	62	Y6
B	93	I/O	63	U8
B	94	I/O	64	W7
B	95	I/O (A2)	65	V8
B	96	I/O	66	Y7
	97	Ext Vss	67	VSSE
B	98	I/O	68	W8
B	99	I/O (A1)	69	U9
B	100	I/O	70	Y8
B	101	I/O (A0)	71	V9
B	102	I/O	72	W9
	103	5V Ext Vdd	73	VDDE
B	104	I/O	74	Y9
B	105	I/O	75	U10
B	106	I/O (D7)	76	W10
B	107	I/O	77	V10
B	108	I/O Clk	78	Y10
	109	Int Vss	79	VSSI
	110	5V Int Vdd	80	VDDI

Edge	Pad	Pad Type	Pin Location	
			DK Suffix 208–Pin QFP	HV Suffix 299–Pin PGA
B	111	I/O Clk	81	V11
B	112	I/O	82	Y11
B	113	I/O	83	U11
B	114	I/O (D6)	84	W11
B	115	I/O	85	W12
	116	Ext Vss	86	VSSE
B	117	I/O (D5)	87	V12
B	118	I/O	88	Y12
B	119	I/O (D4)	89	U12
B	120	I/O	90	Y13
B	121	I/O (D3)	91	W13
	122	5V Ext Vdd		VDDE
B	123	I/O	92	V13
B	124	I/O	93	Y14
B	125	I/O (D2)	94	U13
B	126	I/O	95	W14
B	127	I/O	96	V14
	128	Ext Vss	97	VSSE
B	129	I/O	98	U14
B	130	I/O		Y15
B	131	I/O (D1)	99	V15
B	132	I/O		Y16
B	133	I/O	100	U15
	134	Ext Vss		VSSE
B	135	I/O		V16
B	136	I/O	101	Y17
B	137	I/O		V17
B	138	I/O	102	Y18
B	139	I/O		V18
	140	MODE[0]	103	Y19
	141	Ext Vss		VSSE
	142	MODE[1]	104	W19
	143	3V Ext Vdd		T16
	144	5V Ext Vdd		VDDE
	145	Probe Pad	NOT BONDED	
	146	Ext Vss		VSSE
	147	5V Ext Vdd	105	VDDE
	148	MODE[2]	106	V19

Shaded areas indicate Alternate I/O Zones.

Pinouts for MPA1100 (continued)

Edge	Pad	Pad Type	Pin Location	
			DK Suffix 208–Pin QFP	HV Suffix 299–Pin PGA
	149	5V Int Vdd		U17
	150	MODE[3]	107	W20
	151	Vpp		U18
	152	Clk	108	V20
	153	3V Ext Vdd	109	R16
	154	Ext Vss	110	VSSE
R	155	I/O		T17
R	156	I/O		U20
R	157	I/O (DCLK)	111	T18
R	158	I/O		T19
R	159	I/O	112	R17
	160	Ext Vss		VSSE
R	161	I/O	113	R18
R	162	I/O	114	T20
R	163	I/O		R19
R	164	I/O (D0)	115	R20
R	165	I/O		P17
	166	Ext Vss		VSSE
R	167	I/O	116	P18
R	168	I/O	117	P19
R	169	I/O	118	N17
R	170	I/O	119	P20
R	171	I/O (TDO)	120	N18
	172	Ext Vss	121	VSSE
R	173	I/O (TDI)	122	N19
R	174	I/O	123	N20
R	175	I/O	124	M17
R	176	I/O (TMS)	125	M20
R	177	I/O	126	M18
	178	5V Ext Vdd	127	VDDE
R	179	I/O	128	M19
R	180	I/O (TRSTB)	129	L19
R	181	I/O	130	L17
R	182	I/O	131	L20
R	183	I/O Clk	132	L18
	184	Int Vss	133	VSSI
	185	5V Int Vdd	134	VDDI
R	186	I/O Clk	135	K20

Edge	Pad	Pad Type	Pin Location	
			DK Suffix 208–Pin QFP	HV Suffix 299–Pin PGA
R	187	I/O	136	K18
R	188	I/O	137	K19
R	189	I/O	138	K17
R	190	I/O	139	J20
	191	Ext Vss	140	VSSE
R	192	I/O	141	H20
R	193	I/O	142	J18
R	194	I/O (TCK)	143	H19
R	195	I/O	144	J17
R	196	I/O	145	G20
	197	5V Ext Vdd	146	VDDE
R	198	I/O	147	G19
R	199	I/O	148	H18
R	200	I/O	149	F20
R	201	I/O	150	H17
R	202	I/O	151	F19
	203	Ext Vss		VSSE
R	204	I/O	152	E20
R	205	I/O		G17
R	206	I/O	153	E19
R	207	I/O		F18
R	208	I/O	154	D20
	209	Ext Vss		VSSE
R	210	I/O	155	C20
R	211	I/O		E18
R	212	I/O		B20
R	213	I/O	156	D18
R	214	I/O		C19
	215	Ext Vss		VSSE
	216	3V Ext Vdd		F16
	217	Ext Vss		VSSE
	218	5V Ext Vdd		VDDE
	219	5V Int Vdd		VDDI
	220	Int Vss		VSSI
	221	Ext Vss	157	VSSE
	222	5V Ext Vdd		VDDE
	223	3V Ext Vdd	158	E15
	224	Ext Vss		VSSE

Shaded areas indicate Alternate I/O Zones.

Pinouts for MPA1100 (continued)

Edge	Pad	Pad Type	Pin Location DK Suffix 208–Pin QFP	 HV Suffix 299–Pin PGA
T	225	I/O		C18
T	226	I/O	159	B19
T	227	I/O		C17
T	228	I/O	160	A19
T	229	I/O		C16
	230	Ext Vss		VSSE
T	231	I/O	161	D15
T	232	I/O		A18
T	233	I/O	162	C15
T	234	I/O		A17
T	235	I/O	163	D14
	236	Ext Vss	164	VSSE
T	237	I/O	165	C14
T	238	I/O	166	A16
T	239	I/O	167	D13
T	240	I/O	168	A15
T	241	I/O	169	C13
	242	5V Ext Vdd	170	VDDE
T	243	I/O	171	B13
T	244	I/O	172	A14
T	245	I/O	173	D12
T	246	I/O	174	A13
T	247	I/O	175	C12
	248	Ext Vss	176	VSSE
T	249	I/O	177	B12
T	250	I/O	178	A12
T	251	I/O	179	D11
T	252	I/O	180	A11
T	253	I/O Clk	181	C11
	254	5V Int Vdd	182	VDDI
	255	Int Vss	183	VSSI
T	256	I/O Clk	184	A10
T	257	I/O	185	C10
T	258	I/O	186	B10
T	259	I/O	187	D10
T	260	I/O	188	A9
	261	5V Ext Vdd	189	VDDE

Shaded areas indicate Alternate I/O Zones.

Edge	Pad	Pad Type	Pin Location DK Suffix 208–Pin QFP	 HV Suffix 299–Pin PGA
T	262	I/O	190	B9
T	263	I/O	191	C9
T	264	I/O	192	A8
T	265	I/O	193	D9
T	266	I/O	194	B8
	267	Ext Vss	195	VSSE
T	268	I/O	196	A7
T	269	I/O	197	C8
T	270	I/O	198	B7
T	271	I/O	199	D8
T	272	I/O	200	A6
	273	5V Ext Vdd		VDDE
T	274	I/O	201	A5
T	275	I/O		C7
T	276	I/O	202	B5
T	277	I/O		C6
T	278	I/O	203	A4
	279	Ext Vss		VSSE
T	280	I/O		A3
T	281	I/O (A17)	204	C5
T	282	I/O		A2
T	283	I/O		C3
T	284	I/O	205	B2
	285	Ext Vss		VSSE
	286	5V Ext Vdd		VDDE
	287	Ext Vss	206	VSSE
	288	3V Ext Vdd	207	E6
	289	Int Vss	208	VSSI

299 PGA NOTES:

VSSE Plane:	A1, A20, B3, B6, B14, B16, B18, C4, D2, D5, D7, D16, D19, E4, E8, E13, E17, G3, G18, H5, H16, J2, J19, N5, N16, T5, T8, T13, U2, U6, U16, U19, V4, W3, W16, W18, Y1, Y20
VSSI Plane:	E10, E12, J16, K5, L16, M5, T10, T12
VDDE Plane:	B4, B11, B15, B17, D4, D6, D17, E5, E7, E14, E16, F4, F17, G5, G16, P5, P16, T7, T14, U4, W4, W6, W15, W17
VDDI Plane:	E9, E11, J5, K16, L5, M16, T9, T11

MPA1000 Electrical Specifications

Absolute Maximum Ratings*

Symbol	Parameter	Min	Max	Unit
V_{dd}, V_{ddo}	DC Supply Voltage	–0.5	6.5	V
V_{out}	DC Output Voltage	–0.5	V_{DD} + 0.5	V
V_{in}	DC Input Voltage	–0.5	V_{DD} + 0.5	V
I	DC Current Drain per Pin, Any Single Input or Output		50	mA
T_A	Operating Temperature Range (In Free Air) Commercial Industrial	0 –40	70 85	°C
T_{stg}	Storage Temperature Range	–65	150	°C

* Maximum Ratings are those values beyond which damage to the device may occur. Functional operation should be restricted to the Recommended Operating Conditions. This device contains circuitry to protect the inputs against damage due to high static voltages or electric fields; however, it is advised that normal precautions be taken to avoid application of any voltage higher than maximum rated voltages to this high impedance circuit.

Recommended Operating Conditions

Symbol	Parameter	Min	Max	Unit
V_{dd}	DC Supply Voltage Commercial Industrial	4.75 4.50	5.25 5.50	V
V_{ddo}	Output Supply Voltage 5V Output Supply (5V ext) 3V Output Supply, No I/Os Programmed to 3V 3V Output Supply, 1 or more I/Os Programmed to 3V	V_{dd} V_{dd} 3.0	V_{dd} V_{dd} 3.6	V
V_{IH}	High Level Input Voltage TTL CMOS	2.0 70	V_{dd} 100	V %V_{ddo}
V_{IL}	Low Level Input Voltage TTL CMOS	0 0	0.8 20	V %V_{ddo}

DC Electrical Characteristics – Inputs

Symbol	Parameter	Min	Max	Unit
V_{IH}	High Level Input Voltage TTL CMOS	2.0 70	V_{dd} 100	V %V_{ddo}
V_{IL}	Low Level Input Voltage TTL CMOS	0 0	0.8 20	V %V_{ddo}
I_{IN}	Input Leakage Current	–10	10	µA
I_{OZ}	3–State Leakage Current	–10	10	µA
I_{PD}	Pad Pull–Down (When Selected) at $V_{in} = V_{ddo}$	40	110	µA
I_{PU}	Pad Pull–Up (When Selected) at V_{in} = 0V	20	200	µA
C_{IN}	Input Capacitance (Sample Tested) PGA Package Plastic Packages		15 10	pF

DC Electrical Characteristics – Outputs (4.5 < 5_Ext_Vdd < 5.5; –40°C < T_A < +85°C; V_{dd} = 5V; 3.0 < 3V_Ext_V_{dd} < 3.6)

Symbol	Parameter	Min	Max	Unit
V_{OH}	High Level Output Voltage DPLD_OPLEVEL=5V; DPLD_OPDRIVE=high; I_{OH}=6mA DPLD_OPLEVEL=5V; DPLD_OPDRIVE=low; I_{OH}=4mA DPLD_OPLEVEL=3V; DPLD_OPDRIVE=high; I_{OH}=6mA DPLD_OPLEVEL=3V; DPLD_OPDRIVE=low; I_{OH}=4mA (Note 6.) DPLD_OPLEVEL=5V; DPLD_OPDRIVE=high; I_{OH}=44mA	2.4 2.4 2.1 2.1 2.2		V
V_{OL}	High Level Output Voltage DPLD_OPLEVEL=5V; DPLD_OPDRIVE=high; I_{OH}=–6mA DPLD_OPLEVEL=5V; DPLD_OPDRIVE=low; I_{OH}=–4mA DPLD_OPLEVEL=3V; DPLD_OPDRIVE=high; I_{OH}=–6mA DPLD_OPLEVEL=3V; DPLD_OPDRIVE=low; I_{OH}=–4mA (Note 6.) DPLD_OPLEVEL=5V; DPLD_OPDRIVE=high; I_{OH}=–95mA		0.4 0.4 0.4 0.4 1.4	V

6. Not available on all family members. Please ask your sales representative for more information.

MPA1000 Primary Clock Characteristics

Symbol	Parameter	Min	Typ	Max	Unit
T_{ckin}	Primary Clock Pad to Register Delay		5.6		ns
T_{cks}	Primary Clock Skew			1.0	ns
T_{cwh}	Clock High Time		2.0		ns
T_{cwl}	Clock Low Time		2.0		ns

MPA1000 JTAG Clock Characteristics

Symbol	Parameter	Min	Typ	Max	Unit
F_{cjtag}	Shift Clock Frequency			16	MHz

AC Characteristics (These are Maximum values based on worst case operating points for an MPA1036.)

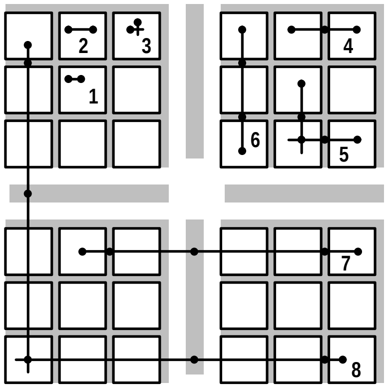
Symbol	Parameter	Speed Grade						Unit
		2I	4I	6I	2	4	6	
T_{cn}	Core Cell AND (Note 7.)	1.1	1.3	1.4	1.0	1.2	1.3	ns
T_{cx}	Core Cell XOR (Note 7.)	1.7	2.0	2.2	1.6	1.8	2.1	ns
T_{ip}	Input Pad Delay	2.7	3.1	3.5	2.6	2.9	3.3	ns
T_{li}	Local Interconnect (Note 8.)	NEG	NEG	NEG	NEG	NEG	NEG	ns
T_{mb}	Medum Bus	0.8	0.9	1.0	0.8	0.9	1.0	ns
T_{mbt}	Medium Bus Turn	2.1	2.4	2.7	2.0	2.2	2.5	ns
T_{g}	Global Bus (Same Quadrant)	1.4	1.7	1.9	1.4	1.6	1.8	ns
T_{gg}	Global Bus (Adjacent Quadrant)	3.3	3.8	4.3	3.2	3.6	4.0	ns
T_{xt}	X–Bus Turn	1.4	1.6	1.8	1.3	1.5	1.7	ns
T_{pb}	Peripheral Bus (Generic)	6.2	7.2	8.1	5.9	6.7	7.6	ns
T_{iwo}	Internal Wired–OR (Full Device)	8.1	9.5	10.6	7.8	8.8	10.0	ns

7. Includes routing.
8. Value is negligible – value is lumped into core cell delays.

Example Delay Paths (These are Maximum values based on worst case operating points for an MPA1036.)

Path	Parameter	Speed Grade						Unit
		2I	4I	6I	2	4	6	
Path 1	Local	1.1	1.3	1.4	1.0	1.2	1.3	ns
Path 2	Medium	1.9	2.2	2.4	1.8	2.1	2.3	ns
Path 3	Medium Turn	3.1	3.6	4.0	3.0	3.5	3.8	ns
Path 4	M–Port–M	3.0	3.5	3.9	2.9	3.4	3.7	ns
Path 5	M–Xturn–M	6.1	7.1	7.7	5.9	6.6	7.4	ns
Path 6	M–G–M	5.9	6.8	7.6	5.6	6.5	7.2	ns
Path 7	M–G–IQ–G–M	7.6	8.8	9.8	7.2	8.3	9.3	ns
Path 8	M–G–IQ–G–Xturn–G–IQ–G–M	14.4	16.8	18.8	13.8	15.8	17.8	ns

M = Medium; G = Global; IQ = Inter–Quadrant Switch. All paths include 1 cell delay. Path is from cell input to next cell input.



MOTOROLA
SEMICONDUCTOR TECHNICAL DATA

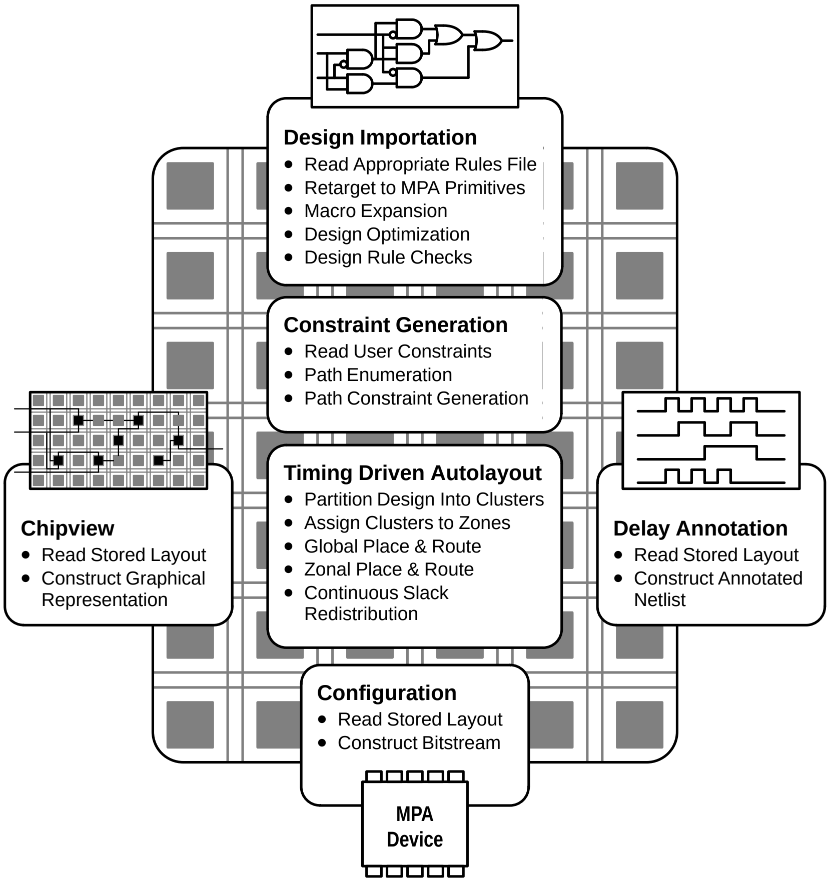
MPA1000 Design System Product Description

Overview

The Motorola Programmable Array (MPA) design system is a bridge between a design capture environment and Motorola field programmable arrays. The MPA design system automatically transforms designs into device configurations which, when loaded into an MPA device, realize a design. A design is automatically analyzed, optimized, transformed into MPA cells, partitioned, placed and routed based on timing constraints for every path in the design. MPA design tools understand and optimally utilize the MPA device architecture; this eliminates the need to learn a new set of rules and makes these tools ideally suited for use with logic synthesis. Full incremental design support reduces design implementation time and powerful library retargeting capabilities allow you to reuse designs which may have been implemented on less capable devices. The MPA design system operates on existing hardware platforms and supports design capture and simulation tools from more than 10 vendors. All these features plus on–line, hypermedia, help make the MPA design system a powerful yet extremely easy to use design implementation engine.

Features

- Push Button Implementation
- Optimal Use of MPA Device Resources
- Optimal Results with Gate Level Design Input
- Library of Common MSI Functions
- Design Flow Manager
- Design Retargeter
- Timing Driven with Integrated Static Timing Analysis
- Layout Delay extraction for post layout simulation
- Layout viewer
- Incremental design support
- On–line, hypermedia, documentation
- Supports all popular design capture and simulation tools
- Lowest cost FPGA development systems.
- Instant access; Downloading via the internet (WWW, ftp).



Push Button Design Implementation

The MPA design system minimizes training investment and automatically generates design implementations which meet timing constraints.

The gate level logic and abundant hierarchical routing resources of the MPA device present a rich implementation media for design implementation. MPA design tools understand and optimally utilize the MPA device resources so there are no elaborate rules to learn or design modifications required to begin design capture. Staying focused on end product design rather than implementation tools or device architecture gets the design done faster and, unlike other programmable solutions, without programmable logic device specificity to impede future design migration efforts. The combination of automatic tools and gate level architecture is ideal for traditional schematic driven or high level language based design capture methods. In fact, logic synthesis tools were originally designed for and produce the most efficient results for targeting gate level devices.

A design is analyzed, optimized, transformed into MPA cells, partitioned, placed and routed based on timing constraints for all paths in the design – automatically. A netlist from one of the popular design capture systems or an existing LPM netlist is imported into the MPA design system. The logic is mapped to a series of MPA cells and the entire resulting netlist is optimized and checked. Based on a simple clock specification, the MPA design system generates timing constraints for all paths in the design. During automatic partitioning, placement and routing path slack time is constantly redistributed insuring only the resources required to meet timing requirements are consumed. Because MPA tools implement the design according to constraints, tool induced design iterations are virtually eliminated. Completed layouts can be transformed into device configurations, as well as annotated simulation netlists. A layout browser is also available.

The MPA design system also includes complete on–line, hypermedia, help covers the device, the design system and the integration kits. Integration kits for Viewlogic, Exemplar, Synopsys, VeriBest, Verilog–SDF, VITAL–SDF, VHDL (1076), Verilog (OVI) and OrCAD are included (contact your vendor for additional kits).All these features add up to a powerful yet extremely easy to use design implementation engine for the MPA product family.

Design Importation

Designs can be captured using schematics, a high level language, or a combination of these entry methods using commercially available design capture and logic synthesis software and the appropriate interface kit. Alternatively, existing designs can be retargeted from other programmable logic devices to the MPA device using commercial logic synthesis tools or the powerful retargetting capabilities provided with MPA design system.

Design importation begins with a netlist and an optional clock specification file. The clock specification file provides a mechanism for the user or design capture tools to document system level timing requirements. In addition, a rich set of attributes can be attached to specific components or nets within the design to specify timing and design pinout constraints.

A retargetting rules file is read and the input netlist is transformed into a series of MPA cells and associated interconnections. Rules files provide a mechanism to perform attribute mapping, cell mapping and macro expansion. By creating custom rule files, the user can extend the importation process from arbitrary sources. The MPA design system comes with rules for it’s native library/EDIF. The resulting netlist is optimized to clip unused logic and remove redundant logic. For example: each MPA cell has programmable input inversion capability. All Inverters or non–inverting buffers can be removed from the netlist and replaced with signal sense information attached to each input.

A series of design rule checks are performed to insure design integrity before the layout process begins.

Constraint Generation

Timing constraints, the optimized MPA netlist and static timing analysis is used to generate path slack constraints for all paths in the design. Each unique signal pathway between a register output and a register input throughout the design are enumerated. The total logic and estimated or real wire delays along the path are summed. The time between the active upstream register clock edge and the next active downstream clock edge minus the downstream register setup time is subtracted from the total path delay. This difference is called path slack. If any path in the design has a negative slack value, the implementation will not function at the required clock rate(s).

Path constraints are utilized throughout the layout process to insure that a design implementation which meets timing constraints is automatically generated. If no clock or timing specifications are provided, the MPA design system uses the fastest possible clock based on very small net delay estimates to generate the path constraints. This usually results in the best possible implementation, but may take longer than the time required to generate a satisfactory rather than best possible result.

Contrast this to other programmable logic design tools which only provide manual net constraint annotation or net criticality assignment. In these cases significant effort is necessary to generate constraints and many costly iterations are required to tune these constraints for a given design. If any changes are made to the design, another costly round of iterations is required.

Autolayout

The autolayout process makes use of the hierarchical organization of the MPA device to minimize run time and deliver implementations that meet timing requirements. Designs which have diverse timing requirements are ideally implemented because path slack estimates are refined throughout the autolayout process insuring only the resources required to meet timing requirements are consumed.

The process begins by flattening the design and partitioning it into small component groups of approximately the same size called clusters. A cluster boundary delay estimation is applied to pull the most tightly constrained paths into a minimum number of clusters. The clusters are then assigned to zones talking into account zonal boundary delay cost and relative zone placement delay costs. Other costs like total number of port connections per zone and are also considered. As assignment proceeds, cluster and zone boundary delay costs are added to each path and slack is recomputed.

Next global placement and routing is done. Global routes begin and end on either I/O cells or port cells. Intrazone placement and routing is deferred to a later phase. During global routing all the port cell and I/O cell locations are fixed and the connections between them established. High fanout nets are constructed in a highly regular manner to insure efficient resource utilization. As in partitioning, slack estimates are refined throughout global routing.

Finally the intrazonal placement and routing is done. Cells assigned to a particular zone are placed and routed to other zone cells or zone port cells. Port cells and core cells are constructed to allow port swapping. Core cells can be routed through if necessary. Allowing core cells to act as routing cells allows dynamic adjustment of routing resources within the zone. Dynamic resource adjustment is a powerful design specific adaptation mechanism.

This process produces a layout from which device configurations, delay back annotations, and chipviews can be generated.

Incremental Design Support

When specification changes necessitate design iterations, simply push the button again. Constraints are automatically recalculated and autolayout only reworks those portions of the design which have changed. Full incremental design support means simple design changes to facilitate design verification can be made quickly and easily.

Delay Back Annotation

Designs can be verified through numerous methods. One particularly useful method is the annotation of device and implementation specific delays back into the original simulation environment to improve system or device level simulation accuracy. A MPA device layout can be transformed into an appropriately formatted delay annotation file or annotated netlist quickly and easily. The annotated delay information represents the worst case delays for a given device speed grade.

Chipview

While the MPA design system provides a rich set of reports describing the implementation of a design, a graphical view of the implementation can be indispensable for reviewing overall layout quality. Chipview provides a graphical view of a completed layout. Chipview can be useful during initial design iterations to visually verify I/O pin placements before commencing PCB layout, for example.

Configuration

A layout can be transformed into a device configuration which, when loaded into the appropriate MPA device, produces a physical design realization. Many formatting options are available. The MPA download pod can be used to emulate a serial PROM. Using the pod, device configuration files can be downloaded to a device directly from the PC or workstation development environment.

Integration Kits

The MPA design system can be used with a large number of commercial electronic design automation software. The Vendor Software List on page 2–56 shows the currently supported vendors and tools. For each supported vendor, an integration kit is provided which facilitates MPA design within that vendors’ environment. Many of these kits are available from Motorola and included at no charge on the MPA design system CD–ROM. Other kits can be acquired directly from the vendor. Refer to the MPA Design System Product List for more information.

Low Cost, Easy Access

MPA Design systems are easy to use, competitively priced and widely available. Copies of MPA design system software supporting 1016 and 1036 can be downloaded from the World Wide Web (WWW) @ http://sps.motorola.com/fpga. Complete kits including download pod, evaluation board, MPA device, CD–ROM and documentation can be ordered from your local authorized Motorola distributor or Motorola sales representative (see Motorola Distributor and Worldwide Sales Office listings in NO TAG on page NO TAG).

Fast, Efficient Design Implentation With Minimal Investment.

That’s MPA!

Vendor Software List

Vendor	Package, Revision	Synthesis	Schematic	Simulation	Timing Analysis
Viewlogic	Workview Office, 7.31 (7.4 Aug 97)	Q2–97	Yes	Yes	Q3–97
Viewlogic	Workview Office, 7.12	Q2–97	Yes	Yes	No
Synopsys	Design Compiler, 3.1	Yes	Yes (Generated)	Yes	No
Exemplar	Galileo, 3.2.5 (4.1 Aug 97)	Yes	Yes (Generated)	Yes	No
Exemplar	Leonardo 4.0.3 (4.1 Aug 97)	Yes	Yes (Generated)	Yes	No
Model Tech	3Q97	No	No	Yes	Yes
Data I/O	Synario, 3.0	Yes	Yes	Yes	No
Cadence	FPGA Designer, TBD	No	No	Yes	Yes
OrCAD	Capture, 7.0	No	Yes	Yes	No
OrCAD	Express, TBD	No	Yes	Yes	No
Protel	Advanced Schematics, 3.23	No	Yes	No	No
VeriBest	VeriBest, VB97	Yes	Yes	Yes	No
Mentor Graphics	Design Architect, A3	Q2–97	Q2–97	Q2–97	Q2–97

Design System Product List

Part Number	Description	Platform		MPADS CD–ROM	Supports 1016/1036	Supports All MPAs	Eval Board	POD	Maintenance	S.R.P.
		PC	WS							
MPA1E/P	Entry Level Kit	X		X	X		X	X		$295
MPA1E/W	Entry Level Kit		X	X	X		X	X		$595
MPA1S/P	Standard Level Kit	X		X		X	X	X	1 Year	$2,295
MPA1S/W	Standard Level Kit		X	X		X	X	X	1 Year	$3,995
MPA1CD/P	Design Software CD	X		X	X					$12.95
MPA1CD/W	Design Software CD		X	X	X					$295
MPA1/POD	Download Pod	X	X			X		X		$195
MPA1/BRD	Evaluation Board	X	X		X (FN)		X			$145
MPA1M12P	Maintenance	X							1 Year	$595
MPA1M12W	Maintenance		X						1 Year	$795

Motorola Programmable Design System Descriptions

MPA1CD – Motorola Design System Software. Single CD–ROM containing MPADS software for both PC and workstation including MPA databook, application notes and all supported EDA vendor integration kits. MPA1CD/D is available from the Motorola Literature Distribution Center – Call 1–800–441–2447 or 303–675–2140. MPA1CD and software downloaded from our web site do not include maintenance.

MPA1E/P – PC Entry Kit. MPADS software with full support for MPA1016 and MPA1036 devices, download pod (MPA1/POD) and an evaluation board (MPA1/BRD), with an 84–pin MPA device, and a CD–ROM, at an attractive price.

MPA1S/P – PC Standard Kit. MPADS software with support for all MPA devices, maintenance for 1 year, download pod (MPA1/POD) and evaluation board (MPA1/BRD).

MPA1E/W – Workstation Entry Kit. MPA Entry kit for workstations – Sun OS V 4.1 (Now), HPUX V 8 (2Q/97).

MPA1S/W – Workstation Standard Kit. MPA standard kit for workstations.

MPA1/POD – Serial port download cable for both workstation and PC. Downloads configuration directly to a MPA device.

MPA1/BRD – Evaluation board including EPROM socket and an MPA1000 device in the 84–pin PLCC package.

MPA1M12P, MPA1M12W – 1 year maintenance (Standard Level Kit includes 1 year of maintenance).

Motorola Integrated Design System Packages for Windows ‘95/NT			
Kit Type	**Part Number**	**Description**	**Suggested Retail Price**
Basic	MPA1WV/BSC	• Schematic Entry & Simulation • MPA Design System Software (MPADS) for MPA1016, MPA1036 • Note: Maintenance Is Not Included	$749
Basic Plus	MPA1WV/BSCPL	• Schematic Entry, Simulation & VHDL Editor/Compiler • MPA Design System Software (MPADS) for MPA1016, MPA1036 • Note: Maintenance Is Not Included	$1,049
Standard	MPA1WV/STD	• Schematic Entry, Simulation & VHDL Editor/Compiler • MPA Design System Software (MPADS) for All MPA1000 Devices • One Year of Maintenance on All Software • Download Pod and Evaluation Board	$3,149
Deluxe	MPA1WV/DLX	• Schematic Entry, VHDL Editor/Compiler, Speedwave VHDL Simulator, Mixed Mode VHDL/Gate Level Simulator • MPA Design System Software (MPADS) for All MPA1000 Devices • One Year of Maintenance on All Software • Download Pod and Evaluation Board	$5,149

For additional information on Workview Office, visit the Viewlogic Web page at: http://www.viewlogic.com/products

The Motorola Integrated Design System incorporates Viewlogic’s Workview Office Tool Suite with Motorola’s Programmable Array Design System (MPADS) providing an integrated, easy to use, complete design environment for MPA1000 FPGAs. Support for other popular design capture and simulation tools, as well as stand–alone MPADS kits and accessories, are also available.

The Motorola Integrated Design System includes:

- Hierarchical Schematic Entry
- Gate Level Simulation
- Simulation Waveform Viewing
- VHDL and Mixed Mode Simulation
- VHDL Entry and VHDL Compilation
- Schematic Generation
- EDIF Netlist Writer
- Design Optimization
- Automatic, Timing Driven, Layout
- Layout Viewing
- Configuration Generation
- Download Hardware and Demo Board

A 30–day evaluation copy of the Integrated Design System with MPA1016 and MPA1036 support is available. Consult factory.