Appendix O

M5 (MACH5) Family Support Supplement

Introduction

AMD’s MACH5xx architecture represents a departure from previous MACH (MACH1xx/2xx/3xx/4xx) families. The earlier MACH families have extremely predictable timing because all signals follow the same paths through internal matrices. The MACH 1 and 2 families have a single internal matrix through which all input signals are routed to PAL blocks. The MACH 3 and 4 families extend internal routability by using Input, Central and Output matrices (see Figure O-1.)

The M5 is a new architecture with a hierarchical interconnect system that provides internal routability of 100%. It also allows you to send signals directly to the PAL blocks without going through the interconnect matrices (as long as equations have fewer than 32 pterms.) Even if functions have more than 32 pterms, the Block Interconnect (see Figure O-2) can connect the signals to another PAL block within the segment. When equations become too large to fit into a segment (4 PAL blocks), they can be routed to other segments via the Segment Interconnect. This means that \( t_{pd} \) becomes longer with increased equation size, with boundaries at 16 and 32 pterms. However, this increased \( t_{pd} \) is offset by the fact that the M5 can connect any two signals internally. This lessens the necessity of having to use up I/O pins to connect signals and eliminates some of the timing problems created by going "off chip". Thus the M5 can make refitting much easier. The major differences between MACH1xx/2xx/3xx/4xx and MACH5 are shown graphically in the following two figures.

**Figure O-1. Simplified MACH 1xx/2xx and 3xx/4xx Block Diagrams**

*Note: Diagram is greatly simplified for illustration. For actual block diagrams, refer to the AMD MACH 1,2,3, and 4 Family Data Book.*
Timing paths are the same for all signals, making for very predictable propagation delays.

To maintain timing paths, all I/Os go through a Switch Matrix (MACH 1 and 2) or an Input Switch Matrix (MACH 3 and 4). There are no direct paths from the outside world (except clocks) to the macrocells.

Highly configurable, due to use of internal routing matrices.

**MACH5xx**

While timing paths are not all the same for the M5 (unlike previous MACH families), this less predictable nature has significant advantages. For example, a block with bonded-out I/O pins could be used as a fast PLD for timing-critical signals. These signals need not go through the internal interconnect matrices.

The use of hierarchical interconnect matrices in the M5 yields increased internal routability (up to 100%). To say it another way, any two signals in an M5 can be connected via the interconnect matrices.
PIL Constructs for M5

The following is a list of PIL (Physical Information Language) properties unique to the M5 family.

Table O-1. PIL Properties for M5 Devices

<table>
<thead>
<tr>
<th>CLOCKED_ENABLED_BY</th>
<th>FORCE_LOCAL_FB</th>
</tr>
</thead>
<tbody>
<tr>
<td>NEG_EDGE</td>
<td>LOCAL_TOGGLE_FEEDBACK</td>
</tr>
<tr>
<td>POS_EDGE</td>
<td>POWER</td>
</tr>
<tr>
<td>CLOCKED_BY_BOTH_EDGES</td>
<td>SLEW_RATE</td>
</tr>
</tbody>
</table>

Segment and Block Assignments

The SECTION construct and the TARGET statement are used to specify how signals are routed in a Segment and Block of an M5 device.

Syntax

TARGET S<seg_id>[B<block_id>]

Where:

seg_id is an optional segment identifier from 0..7(for M5)
block_id is an optional block identifier from a..d

You can specify just the targeted segment with:

TARGET 'S0'; "section targeted at segment 0"
or specify both targeted segment and block with:

TARGET 'S0Ba'; "section targeted at segment 0, block A"

Example

DEVICE
  TARGET ‘TEMPLATE MV256_68 QFP-100-M256’; "place group into MV256"

SECTION
  TARGET ‘S1’; "force q1 into MACH5 segment 1 on pin #8"
  q1:8;
END SECTION;

SECTION
  TARGET ‘S0Ba’; "force out7..out8 into MACH5 segment 0 block A"
  ...
  out7:5, out8:6; out9: out10: S0Bam2: "assignment with physical pin or node numbers"
END SECTION;
END DEVICE;
**Pin and Node Assignment**

An M5 device has both physical (or absolute) pins (the ones on the device package) and relative node numbers (i.e., node locations within the device.) For each node number there is a corresponding node name (which is generally easier to remember than the node number.) Using node numbers instead of absolute pin numbers may eliminate the need to alter the .pi file if you go to another device.

There are two feedbacks associated with each macrocell in an M5:

- **Macrocell feedback** which feeds back immediately after the macrocell, to the Block Interconnect.

- **Pin feedback** which feeds back after the tristate buffer. The pin feedback may or may not bond-out to a physical pin.

![Figure O-3. Simplified Macrocell and Pin Feedback for M5](image_url)

**Syntax**

```
[S <seg_id>][<B <block_id>] <feedback_id>
```

Where:

- `seg_id` is an optional segment identifier from 0..7 (for M5)
- `block_id` is an optional block identifier from a..d
- `feedback_id = M<mcell_no> | P<mcell_no>`
  - `mcell_no` is the macrocell number from 0..16
  - `M<mcell_no>` is the macrocell feedback
  - `P<mcell_no>` is the pin feedback.

Note that there is a node number for every pin feedback, regardless of whether or not the pin feedback is bonded-out to a physical pin. If the pin feedback is bonded-out, there is also a corresponding absolute pin number. If
you specify absolute pin numbers, they will be reproduced in all output files of the fitter.

The following table defines relative node names and their virtual pin numbers. Virtual pin numbers are defined for internal device use only and are unique for the entire M5 family.

Table O-2. M5 Node Names and Pin Numbers

<table>
<thead>
<tr>
<th>Relative Node Names</th>
<th>Virtual Pin Numbers</th>
</tr>
</thead>
<tbody>
<tr>
<td>S0BaM00..S0BaM15</td>
<td>0-15</td>
</tr>
<tr>
<td>S0BaP00..S0BaP15</td>
<td>16-31</td>
</tr>
<tr>
<td>S0BbM00..S0BbM15</td>
<td>32-47</td>
</tr>
<tr>
<td>S0BbP00..S0BbP15</td>
<td>48-63</td>
</tr>
<tr>
<td>S0BcM00..S0BcM15</td>
<td>64-79</td>
</tr>
<tr>
<td>S0BcP00..S0BcP15</td>
<td>80-95</td>
</tr>
<tr>
<td>S0BdM00..S0BdM15</td>
<td>96-111</td>
</tr>
<tr>
<td>S0BdP00..S0BdP15</td>
<td>112-127</td>
</tr>
<tr>
<td>S1BaM00..S1BaM15</td>
<td>128-143</td>
</tr>
<tr>
<td>S1BaP00..S1BaP15</td>
<td>144-159</td>
</tr>
<tr>
<td>S1BbM00..S1BbM15</td>
<td>160-175</td>
</tr>
<tr>
<td>S1BbP00..S1BbP15</td>
<td>176-191</td>
</tr>
<tr>
<td>S1BcM00..S1BcM15</td>
<td>192-207</td>
</tr>
<tr>
<td>S1BcP00..S1BcP15</td>
<td>208-223</td>
</tr>
<tr>
<td>S1BdM00..S1BdM15</td>
<td>224-239</td>
</tr>
<tr>
<td>S1BdP00..S1BdP15</td>
<td>240-255</td>
</tr>
<tr>
<td>S7BaM00..S7BaM15</td>
<td>896-911</td>
</tr>
<tr>
<td>S7BaP00..S7BaP15</td>
<td>912-927</td>
</tr>
<tr>
<td>S7BbM00..S7BbM15</td>
<td>928-943</td>
</tr>
<tr>
<td>S7BbP00..S7BbP15</td>
<td>944-959</td>
</tr>
<tr>
<td>S7BcM00..S7BcM15</td>
<td>960-975</td>
</tr>
<tr>
<td>S7BcP00..S7BcP15</td>
<td>976-991</td>
</tr>
<tr>
<td>S7BdM00..S7BdM15</td>
<td>992-1007</td>
</tr>
<tr>
<td>S7BdP00..S7BdP15</td>
<td>1008-1023</td>
</tr>
</tbody>
</table>

Example

```
INPUT j1:S2BaM00
  "route j1 to macrocell S2BaM00"
```
Unary Assignment

The `.pi` file property UNARY is used to place a signal on an input register or input latch. The UNARY property must be specified on the output signal of the unary function. For more information on unaries, see Chapter 16.

Example

Source File

```plaintext
INPUT ui, iclk;
NODE uo CLOCKED_BY iclk;

uo = ui;
```

Physical Information File

```plaintext
DEVICE
SECTION
  TARGET 'S0Ba';
  INPUT ui;
  NODE uo { UNARY };
END DEVICE;
```

Dual Feedback

Dual feedback is the simultaneous use of both feedback paths, internal and pin (see Figure O-3). There are no DSL or `.pi` constructs for specifying dual feedback. The dual feedback is supported by writing an intermediate node equation and setting the pin feedback equal to node feedback.

The M5 fitter will look for such dual feedback equations and place them on the internal and pin feedback of the same macrocell. There might be exceptions (unknown at this time) that will not allow placement of such dual feedback equations on the same macrocell internal/pin feedback. One possible example is lack of routing resources.

One final note is that the node collapsing in PLOpt will collapse the intermediate node away unless you preserve the intermediate node in the `.pi` file by explicitly declaring it as PHYSICAL.

Example 1

```plaintext
INPUT i1, i2, i3, i4;
OUTPUTS out1, out2, out3;
OUTPUT pin_fb; "pin_fb has to be placed on bonded out pin. If you consider this "wasting an I/O pin, then declare this a node instead.

NODE node_fb;

node_fb = i1 * i2 + i3 * i4; "node feedback equation
pin_fb = node_fb; "intermediate node equation

out1 = i2 * i3 * node_fb; "node feedback used
out2 = i2 + i4 * pin_fb; "pin feedback used
out3 = i2 * node_fb + i3 * pin_fb; "both node and pin feedback on same eqn
```

Example 2

```plaintext
INPUT i1, i2, i3, i4;
OUTPUTS out1, out2, out3;
NODE pin_fb; "pin_fb can be placed on either bonded-out or buried "out pin feedback

NODE node_fb;
```
node_fb = i1 * i2 + i3 * i4; "node feedback equation
pin_fb = node_fb; "intermediate node equation

out1 = i2 * i3 * node_fb; "node feedback used
out2 = i2 + i4 * pin_fb; "pin feedback used
out3 = i2 * node_fb + i3 * pin_fb; "both node and pin feedback on same eqn

The difference between the two examples above is that the signal \( pin_fb \) is an OUTPUT in example 1 and a NODE in example 2. The M5 fitter for PLDSynthesis II supports:

- Pin feedback and internal feedback nodes may be declared as NODES.
- Internal feedback may be declared as a NODE while pin feedback is declared a PIN.

**Forcing the Feedback Path to be Local**

There may be cases, for timing reasons, in which you want all feedbacks contained within the same PAL block. You can do this in the M5 with the FORCE_LOCAL_FB property. This property can be used at the DEVICE, SECTION, or signal level in the .pi file. This prevents any L2 or L3 routing from being used by the fitter.

**Example**

**Source File**

```
INPUT clk, rst, load, up_down, data[7..0] ;
OUTPUT count[7..0] CLOCKED_BY clk RESET_BY rst;

IF load = 0 THEN
  IF up_down = 1 THEN
    count = count + 1;
  ELSE
    count = count - 1;
  END IF;
ELSE
  count = data;
END IF;
```
Physical Information File

"This example shows the use of the FORCE_LOCAL_FB in a GROUP and SECTION. Note that you have to specify "FORCE_LOCAL_FB at the signal level in a GROUP.

DEVICE
  TARGET 'PART_NUMBER AMD M5-256/160-7HC';
  GROUP
    COUNT[7];
    COUNT[6];
    COUNT[5];
    DATA[7]    {FORCE_LOCAL_FB};
    DATA[6]    {FORCE_LOCAL_FB};
  END GROUP;

  SECTION
    TARGET 'S1Bb';
    {FORCE_LOCAL_FB};
    COUNT[3];
    COUNT[2];
    COUNT[1];
    COUNT[0];
    DATA[5];
    DATA[4];
  END SECTION;
END DEVICE;

**Toggle Register Feedback**

A toggle (T) register is implemented by taking the feedback of the register output $Q$ and XORing it with the D- register input. The toggle feedback can be

- A local feedback
- Routed via a level 2 demux and the segment bus (non-local).

The property LOCAL_TOGGLE_FEEDBACK is used to force local toggle feedback.

The LOCAL_TOGGLE_FEEDBACK property can be specified at the device, SECTION or signal (outputs and nodes only) level.

If a local feedback path cannot be found for the toggle feedback, the fitter generates a warning.
Dual-Edge Clocking

The M5 has three clocking options:

- Selectable positive/negative edge clocking
- Clocking on both edges
- Complementary clocking, creating an inverse of clock line 3 (CLK2) on clock line 4 (CLK3).

The .pi file property BOTH_EDGES_OF lets you make use of either or both edges of the specified clock. You can use enables to specify negative or positive edge clocking by means of two keywords: NEG_EDGE_ENABLED_BY and POS_EDGE_ENABLED_BY.

If a BOTH_EDGES_OF statement is used without one of the edge ENABLED_BY statements, the equation defaults to clocking on BOTH edges.

Complementary clocking is available if the macrocell is not controlled by a BOTH_EDGES_OF construct. Complementary clocking uses clock line 3 (CLK2) as the primary clock and clock line 4 (CLK3) as the inverted clock.

Syntax

```
OUTPUT signal_name CLOCKED_BY BOTH_EDGES_OF clk_name
    NEG_EDGE_ENABLED_BY enable_name;
    POS_EDGE_ENABLED_BY enable_name;
```

Examples

```
INPUT clk1, clk2, ce1, ce2;
OUTPUT out1 CLOCKED_BY BOTH_EDGES_OF clk1; *clocks out1 on both edges of clk1
OUTPUT out2 CLOCKED_BY BOTH_EDGES_OF clk2
    POS_EDGE_ENABLED_BY oe1
    NEG_EDGE_ENABLED_BY oe2;
    *clocks out2 on either edge of clk2, determined by enables oe1 and oe2
```

MACH_UTILIZATION

The MACH_UTILIZATION property specifies the amount of reserve capacity to leave available in a device. This affects the use of pterms and macrocells.

Syntax

```
{MACH_UTILIZATION percent} ;
```

Where percent is the percentage of device resources to be used. The range of values is 0 to 100.

The unused resources are distributed throughout the device. There are two reasons to reserve some resources in a device.
1) Resources may be reserved to allow for expansion of logic.

2) Resources may be reserved to ease and speed the fitting process. Simply put, it is easier for PLFit to place and route a solution at 80% utilization than at 100% utilization. If design iteration speed is more important than density (e.g., earlier in the design cycle or for refitting), set the utilization factor to a lower value.

**MAX_PTERMS**

The MAX_PTERMS property, while not specific only to MACH, provides a means of tuning the optimization to best fit a design into MACH parts. The optimization process collapses combinatorial nodes in the design up to a size specified by MAX_PTERMS. The value used for this property affects fitting into MACH parts. If the value is low, the design will typically be implemented as a larger number of smaller equations. This makes placement somewhat easier because smaller functions do not place demand on the pterm allocation mechanism, but more smaller functions may require more routing resources and may require more overall macrocell logic. At the other end, fewer larger functions may ease the routing requirements, but be harder to place, because the demand for pterms may cause conflicts in placing functions together in a PAL block.

For more information on Optimizing parameters, see Chapter 13.

**Suitable Optimizing Parameters for M5 Devices**

For general purposes, the following parameters may be used in the .pi file for designs targeted at MACH5 devices.

```plaintext
{
MAX_PTERMS 16,
MAX_XOR_PTERMS 15,
MAX_SYMBOLS 20,
POLARITY_CONTROL TRUE,
XOR_POLARITY_CONTROL TRUE
}
```

**The Effect of MAX_PTERMS and MAX_XOR_PTERMS**

The exact effect of changing the optimizing parameters can be checked by checking the nodes in the .doc file after optimizing. The number of nodes will generally go down as the MAX_PTERMS parameter goes up.

The effect of changing the parameters is summarized here:

**Higher MAX_PTERMS**

- More node collapsing
- Larger functions
- Faster implementation
- May increase routing requirements
Lower MAX_PTERMS

- Less node collapsing
- Smaller functions
- Slower implementation
- May increase routing requirements

Note that either High or Low MAX_PTERMS can cause greater routing demand.

Lower MAX_PTERMS can produce more internal nodes which must be routed to the equations where they are used.

Higher MAX_PTERMS can allow a node to be collapsed into multiple equations so that the signals required to generate the node may be needed in multiple places. Furthermore, large equations may require large numbers of signals to be routed into the block where the equation is placed, producing a locally high routing demand.

In critical fitting cases, it may be necessary to try several optimizing values to get satisfactory results.

**Power Levels**

The syntax for specifying power level is:

```
POWER LOW | MED_LOW | MED_HIGH | HIGH
```

Power levels can be specified at a signal, SECTION or device level. The fitter will check the power levels for consistency across the various levels. Error messages will be printed out when the power levels specified do not match. If none is specified, the default power level will be HIGH.

```
SECTION
    TARGET 'S1Bb';  "force out7..out8 into MACH5 segment 1 block b
    q2:M00;        "placements are local to block
    INPUT o2:P02;
    f2:M01 {POWER LOW};  "Use LOW POWER setting
END SECTION;
```

**Slew Rates**

The syntax for specifying slew rate is:

```
SLEW_RATE SLOW | FAST
```

Slew rate can be specified for signal, SECTION or device level. The fitter will check the slew rates for consistency across the various levels. If slew rates specified do not match, the fitter will generate an error.

```
SECTION
    TARGET 'S1Bb';  "force out7..out8 into MACH5 segment 1 block b
    q2:M00;        "placements are local to block
    { SLEW_RATE FAST }  
    INPUT o2:P02; { SLEW_RATE SLOW };  "slew_rate is SLOW
END SECTION;
```
There is also a factory-programmed device-level downgrade to SLOW. When set to SLOW, it overrides the FAST slew rate attribute for individual signals. If individual signals are explicitly specified with a FAST slew rate and the device level slew rate has been downgraded to SLOW, the fitter will generate a warning.

**Design Documentation**

**Document File**

The document file (*design_name.doc*) contains information about the various stages of compilation and partitioning. The following information is contained in the *.doc* file:

- Information about the design (title, designer, date, company, etc) and switch values specified for compiler and optimizer functions.
- Explicit (or reduced) design equations that are realized in the final layout.
- A list of the solutions generated for the design.
- Partitioning criteria used in generating the device solutions.
- Pinout diagrams of the device solution selected.
- A list of possible devices for the templates in the solution.
- A wire list.

For more information on these sections, please see Chapter 17.

**Report File**

In addition to the *.doc* file, a report file, *design_name.rpt* will be generated for an M5 device. The report file generally contains the following sections:

**Heading**

This section generally contains the following information:

- Date when the design was run through the fitter.
- Part type and device number
- Package type
- User supplied design information
Example

DATE: Fri Jan 26 14:44:48 1996
DESIGN: prob1.fb
DEVICE: MV256_160:1

**SUMMARY STATISTICS**

This section summarizes the design in terms of number of clocks, inputs, nodes and outputs at the device level and its various sub-partitions namely, segments and PAL blocks. Power levels for each block will be specified here.

Example

**SUMMARY STATISTICS:**

10 Inputs
32 Outputs
0 Tri-states
124 Nodes

Functions by block:

<table>
<thead>
<tr>
<th>Segment</th>
<th>8</th>
<th>7</th>
<th>12</th>
<th>12</th>
</tr>
</thead>
<tbody>
<tr>
<td>S0</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>S1</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>S2</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>S3</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

D Register Macrocells 36
T Register Macrocells 24
D Latch Macrocells 0
Combinatorial Macrocells 92
D Input Registers 0
D Input Latches 0

Xor Equations 24
Single-Pterm Equations 23
Total Pterms Required 867

**POWER RESOURCE UTILIZATION**

The POWER SUMMARY section shows the following:

- Number of blocks with power set to LOW
- Number of blocks with power set to MED_LOW
- Number of blocks with power set to MED_HIGH
- Number of blocks with power set to HIGH
Example

POWER SUMMARY:

Number of blocks with power set to LOW is 0
Number of blocks with power set to MED_LOW is 0
Number of blocks with power set to MED_HIGH is 0
Number of blocks with power set to HIGH is 16

DEVICE RESOURCE UTILIZATION

This section provides utilization statistics for the different device resources at the device, segment and PAL block partitions. A table is provided for each partition with the following columns:

- Resource - Name of resource. The resources available for each block may be different.
- Available - Available resource count for the partition.
- Used - Used resource count for the partition.
- Remaining - Unused resource count for the partition.
- Percent - Percentage resource utilization for the partition.

The resource types referenced in these tables are defined here.

- Clocks - Clock pins used for clock signals
- Pins - Input and I/O pins used in any capacity
- I/O Pins - Number of bonded-out pin feedbacks
- Input Regs - Macrocells used as input registers
- Macrocels - Macrocells without output/buried distinction
- Pterms - AND array rows used in equation generation
- Feedbacks - Inputs to the Switch Matrix
- Fanouts - Inputs to the AND Arrays
- Blk Clocks - Number of selectable clock lines for each block

The resource types for the device and segment partitions are:

- Clocks
- Pins
- Input Regs
The resource types for the PAL block partitions are divided into two groups:

- Clock generator block.
  - Clocks
  - Pterms
  - Blk Clocks

- Macrocell block.
  - I/O Pins
  - Input Regs
  - Macrocells
  - Pterms
  - Feedbacks
  - Fanouts

**Example**

**DEVICE RESOURCE UTILIZATION:**

<table>
<thead>
<tr>
<th>Resource</th>
<th>Available</th>
<th>Used</th>
<th>Remaining</th>
<th>%</th>
</tr>
</thead>
<tbody>
<tr>
<td>DEVICE</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Clock Pins:</td>
<td>4</td>
<td>1</td>
<td>3</td>
<td>25</td>
</tr>
<tr>
<td>I/O Pins:</td>
<td>160</td>
<td>41</td>
<td>119</td>
<td>25</td>
</tr>
<tr>
<td>Input Regs:</td>
<td>32</td>
<td>0</td>
<td>32</td>
<td>0</td>
</tr>
<tr>
<td>Macrocells:</td>
<td>256</td>
<td>156</td>
<td>100</td>
<td>60</td>
</tr>
<tr>
<td>Control Pterms:</td>
<td>144</td>
<td>16</td>
<td>128</td>
<td>11</td>
</tr>
<tr>
<td>Cluster Pterms:</td>
<td>1024</td>
<td>876</td>
<td>148</td>
<td>85</td>
</tr>
<tr>
<td>1-pt Clusters:</td>
<td>256</td>
<td>180</td>
<td>76</td>
<td>70</td>
</tr>
<tr>
<td>3-pt Clusters:</td>
<td>256</td>
<td>252</td>
<td>4</td>
<td>98</td>
</tr>
<tr>
<td>Signal Sources:</td>
<td>512</td>
<td>133</td>
<td>379</td>
<td>25</td>
</tr>
<tr>
<td>Array Inputs:</td>
<td>512</td>
<td>264</td>
<td>248</td>
<td>51</td>
</tr>
<tr>
<td>Intersegment Lines:</td>
<td>128</td>
<td>9</td>
<td>119</td>
<td>7</td>
</tr>
</tbody>
</table>

SEGMENT 0

| Clock Pins:         | 4         | 1    | 3         | 25 |

PARTITION GROUPS

This section shows how functions (outputs and nodes) are assigned to the PAL blocks. It shows which signals must be routed to the PAL block to generate the functions assigned to the block. It also shows how many unique clocks, enables and register preset/reset equations are required for the assigned functions.

Example

PARTITION GROUPS:

Block ‘S0Ba’; Partition 0; Group-type FIXED_GROUP;
1 Clocks; 0 Enables; 1 Register Sets
8 Functions
#prep_4.12.large-0 #prep_4.12.large-1 #prep_4.12.large-2
#prep_4.12.large-3 #prep_4.12.large-4
15 Signals
clk rst q8[7]
q8[0] prep_4.12.large-0 prep_4.12.large-1
prep_4.12.large-2 prep_4.12.large-3 prep_4.12.large-4

Block ‘S0Bb’; Partition 1; Group-type FIXED_GROUP;
SIGNAL DIRECTORY

Clocks, inputs, outputs and nodes on the part are listed with specific assignment information for each signal. Slew rate which is on a per-signal basis on the M5 will also be listed here.

The signal directory table will have the following columns:

- **Signal #** - The index number used to reference the signal
- **Signal Name** - The user identifier for the signal
- **Source Type** - [Input | Hidden | Output | Biput | Internal] with register type qualifiers
- **PalBlk** - Pal Block where output or node is assigned
- **Clusters** - Further subdivided into two columns:
  - **Used** - Number of Pterm Clusters used to generate function
  - **Unused PTs** - Unused Pterms left in used clusters.
- **Pal Block Inputs** - Array input lines for Signal Fanouts
### SIGNAL DIRECTORY:

Notes: Register type suffix ‘_X’ indicates XOR used;
Register type suffix ‘_LT’ indicates function is LOW_TRUE.
‘RS_SWAP’ flags functions which are preset at power-on.
’OE’ flags tri-state functions.

[ 0] Output: O[7]  
Pin 168 (I/O)  Block S3Bd  Macrocell_02  4 Pterm COMB

Pin 169 (I/O)  Block S3Bd  Macrocell_03  2 Pterm COMB

Pin 170 (I/O)  Block S3Bd  Macrocell_04  2 Pterm COMB

Pin 165 (I/O)  Block S3Ba  Macrocell_00  1 Pterm COMB

Pin 171 (I/O)  Block S3Bd  Macrocell_05  2 Pterm COMB

Pin 163 (I/O)  Block S3Ba  Macrocell_02  1 Pterm COMB

Pin 164 (I/O)  Block S3Ba  Macrocell_01  3 Pterm COMB

[ 7] Output: O[0]  
Pin 172 (I/O)  Block S3Bd  Macrocell_06  2 Pterm COMB

[ 8] Node: q1[7]  
S3BcM2  Block S3Bc  Macrocell_02  4 Pterm COMB

S3BcM1  Block S3Bc  Macrocell_01  2 Pterm COMB

S3BdM1  Block S3Bd  Macrocell_01  2 Pterm COMB

S3BdM0  Block S3Bd  Macrocell_00  1 Pterm COMB

[12] Node: q1[3]  
S3BdM12 Block S3Bd  Macrocell_12  2 Pterm COMB
## FANOUT TABLE

The headings in the table have the following meanings:

- **Signal_Src** - Signal name from the SIGNAL DIRECTORY LIST.
- **ISL#** - Intersegment line number to which Signal_Src connects.
- **SL#** - Segment Line Number.
- **Src SL#** - Segment line number for the same segment as the source signal.

### Example

**FANOUT TABLE:**

<table>
<thead>
<tr>
<th>PASS/FAIL</th>
<th>Signal_Src</th>
<th>ISL#</th>
<th>SL#</th>
<th>Blk</th>
<th>SL#</th>
<th>Mux</th>
<th>Blk</th>
<th>SL#</th>
<th>Mux</th>
<th>Blk</th>
<th>SL#</th>
<th>Mux</th>
</tr>
</thead>
<tbody>
<tr>
<td>FAIL</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>PASS</td>
<td>S0BaM3</td>
<td>---</td>
<td>---</td>
<td>S0Ba</td>
<td>66</td>
<td>M17I2</td>
<td>S0Bd</td>
<td>66</td>
<td>M16I3</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>PASS</td>
<td>S0BaM4</td>
<td>---</td>
<td>---</td>
<td>S0Ba</td>
<td>97</td>
<td>M24I0</td>
<td>S0Bd</td>
<td>97</td>
<td>M22I5</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>PASS</td>
<td>S0BaM5</td>
<td>---</td>
<td>---</td>
<td>S0Ba</td>
<td>74</td>
<td>M16I1</td>
<td>S0Bd</td>
<td>74</td>
<td>M18I4</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>PASS</td>
<td>S0BaM8</td>
<td>---</td>
<td>---</td>
<td>S0Ba</td>
<td>75</td>
<td>M27I2</td>
<td>S0Bd</td>
<td>75</td>
<td>M17I4</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>PASS</td>
<td>S0BaM11</td>
<td>---</td>
<td>---</td>
<td>S0Ba</td>
<td>104</td>
<td>M07I0</td>
<td>S0Bd</td>
<td>104</td>
<td>M06I5</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>PASS</td>
<td>S0BaP4</td>
<td>35</td>
<td>116</td>
<td>S0Bc</td>
<td>116</td>
<td>M11I5</td>
<td>S0Bd</td>
<td>116</td>
<td>M29I4</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>S1Bd</td>
<td>119</td>
<td>M05I6</td>
<td>S2Bc</td>
<td>98</td>
<td>M09I5</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>S3Bc</td>
<td>102</td>
<td>M22I5</td>
<td>S3Bd</td>
<td>102</td>
<td>M19I5</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>PASS</td>
<td>S0BaP5</td>
<td>11</td>
<td>117</td>
<td>S0Ba</td>
<td>117</td>
<td>M01I0</td>
<td>S0Bb</td>
<td>117</td>
<td>M13I1</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>S0Bd</td>
<td>117</td>
<td>M03I6</td>
<td>S1Ba</td>
<td>101</td>
<td>M21I3</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>S1Bc</td>
<td>101</td>
<td>M15I5</td>
<td>S1Bd</td>
<td>101</td>
<td>M18I5</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>S2Bb</td>
<td>114</td>
<td>M04I1</td>
<td>S2Bc</td>
<td>114</td>
<td>M24I6</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>S3Bc</td>
<td>100</td>
<td>M29I0</td>
<td>S3Bb</td>
<td>100</td>
<td>M12I1</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>S3Bd</td>
<td>100</td>
<td>M13I5</td>
<td>S3Bc</td>
<td>100</td>
<td>M08I5</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>PASS</td>
<td>S0BaP6</td>
<td>91</td>
<td>86</td>
<td>S0Bc</td>
<td>86</td>
<td>M10I3</td>
<td>S0Bd</td>
<td>86</td>
<td>M04I3</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>S1Bd</td>
<td>87</td>
<td>M28I4</td>
<td>S2Bc</td>
<td>65</td>
<td>M22I3</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>S3Bc</td>
<td>71</td>
<td>M20I3</td>
<td>S3Bd</td>
<td>71</td>
<td>M18I3</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>PASS</td>
<td>S0BaP7</td>
<td>23</td>
<td>119</td>
<td>S0Bc</td>
<td>119</td>
<td>M03I5</td>
<td>S0Bd</td>
<td>119</td>
<td>M05I6</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>S1Bd</td>
<td>99</td>
<td>M16I5</td>
<td>S2Bc</td>
<td>115</td>
<td>M01I5</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>S3Bc</td>
<td>111</td>
<td>M31I5</td>
<td>S3Bd</td>
<td>111</td>
<td>M30I6</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>PASS</td>
<td>S0BaP9</td>
<td>126</td>
<td>89</td>
<td>S0Bc</td>
<td>89</td>
<td>M09I4</td>
<td>S0Bd</td>
<td>89</td>
<td>M07I4</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>S1Bd</td>
<td>73</td>
<td>M23I5</td>
<td>S2Bc</td>
<td>92</td>
<td>M14I5</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>S3Bc</td>
<td>88</td>
<td>M10I4</td>
<td>S3Bd</td>
<td>88</td>
<td>M02I4</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
POWER TABLE

Example

POWER TABLE:

<table>
<thead>
<tr>
<th>SEGMENT 0</th>
<th>SEGMENT 1</th>
<th>SEGMENT 2</th>
<th>SEGMENT 3</th>
</tr>
</thead>
<tbody>
<tr>
<td>HIGH</td>
<td>HIGH</td>
<td>HIGH</td>
<td>HIGH</td>
</tr>
<tr>
<td>HIGH</td>
<td>HIGH</td>
<td>HIGH</td>
<td>HIGH</td>
</tr>
<tr>
<td>HIGH</td>
<td>HIGH</td>
<td>HIGH</td>
<td>HIGH</td>
</tr>
<tr>
<td>HIGH</td>
<td>HIGH</td>
<td>HIGH</td>
<td>HIGH</td>
</tr>
</tbody>
</table>

BLOCK CONFIGURATION TABLES

Example

BLOCK CONFIGURATION TABLES:

Notes: '*' indicates that the pin is bonded-out

BLOCK 'S0Ba': POWER=HIGH

CONTROL PTERMS:

RST0 = rst ;

BLOCK CLOCKS:

BLK_CLK 2 (PIN_CLOCK,POL=HIGH) : clk ;
BLK_CLK 3 (PIN_CLOCK,POL=LOW) : clk ;

ARRAY INPUTS:

| [---] | [165] | [---] | [138] | [139] | [---] | [135] | [155] |
| [140] | [---] | [---] | [---] | [---] | [---] | [---] | [---] |
| [153] | [151] | [---] | [---] | [136] | [134] | [133] | [---] |
| [152] | [---] | [137] | [154] | [---] | [---] | [---] | [---] |

Inputs I0 to I7

Inputs I8 to I15

Inputs I16 to I23

Inputs I24 to I31

Array inputs I0 through I31 are assigned signal names from the SIGNAL DIRECTORY list.

In the following segment, labels have the following meanings:

- **Pterms Used** - number of pterms used on this macrocell. If the column has 1+7, it means 7 pterms were used and one pterm was steered from elsewhere.

- **Pterms Avl** - number of pterms available for this macrocell.

- **PT Map** - indicates whether pterm was applied to the XOR or product term cluster (OR input.)

- **POL** - indicates polarity of the signal.
- **CLK** - indicates which clock from the clock generator was used.
- **Reg Ctrl** - indicates whether signal was combinatorial or registered.
- **Slew** - Slew rate set.
- **OE** - indicates whether the output enable was high or low.
- **Node** - signal name from the SIGNAL DIRECTORY LIST.
- **Pin** - actual pin number.

<table>
<thead>
<tr>
<th>C C C C C C C C C C C C P C</th>
<th>Pterms</th>
<th>PT</th>
<th>O L Reg</th>
<th>Used Avl Map</th>
<th>L K Ctrl</th>
<th>Slew</th>
<th>Node</th>
<th>OE</th>
<th>Pin</th>
</tr>
</thead>
<tbody>
<tr>
<td>00 1</td>
<td>-</td>
<td>-</td>
<td></td>
<td>0 1 2 3 4 5</td>
<td>0 SUM L 3</td>
<td>COMB</td>
<td>FAST</td>
<td>[---] VCC</td>
<td>[120] *</td>
</tr>
<tr>
<td>01 3</td>
<td>-</td>
<td>-</td>
<td></td>
<td>3 0 SUM L 3</td>
<td>COMB FAST</td>
<td>[---] VCC</td>
<td>[123] *</td>
<td></td>
<td></td>
</tr>
<tr>
<td>02 -1</td>
<td>-</td>
<td>-</td>
<td></td>
<td>1 0 SUM L 3</td>
<td>COMB FAST</td>
<td>[---] VCC</td>
<td>[122] *</td>
<td></td>
<td></td>
</tr>
<tr>
<td>03 -4 31</td>
<td>1+7</td>
<td>0</td>
<td>XOR H 2</td>
<td>RST0 SLOW</td>
<td>[151] GND</td>
<td>[---] *</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>04 -3 1 44</td>
<td>1+11</td>
<td>0</td>
<td>XOR H 2</td>
<td>RST0 SLOW</td>
<td>[152] GND</td>
<td>[162] *</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>05 -3 4 4</td>
<td>11</td>
<td>0</td>
<td>SUM H 2</td>
<td>RST0 SLOW</td>
<td>[153] GND</td>
<td>[165] *</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>06 -</td>
<td>-</td>
<td>-</td>
<td></td>
<td></td>
<td>[---] GND</td>
<td>[156] *</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>07 -</td>
<td>-</td>
<td>-</td>
<td></td>
<td></td>
<td>[---] GND</td>
<td>[163] *</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>08 -3 3 4 4</td>
<td>13</td>
<td>1</td>
<td>SUM H 2</td>
<td>RST0 SLOW</td>
<td>[154] GND</td>
<td>[---] *</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>09 -1</td>
<td>-</td>
<td>-</td>
<td></td>
<td></td>
<td>[---] GND</td>
<td>[157] *</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>10 -1</td>
<td>-</td>
<td>-</td>
<td></td>
<td></td>
<td>[---] GND</td>
<td>[---] *</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>11 -4 4 12</td>
<td>0</td>
<td>SUM H 2</td>
<td>RST0 SLOW</td>
<td>[155] GND</td>
<td>[---] *</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Application Note:

Clock Assignments

M5 devices have a special clock assignments at the PAL block level.

The following clock assignments would fail:

1. (>1) dual-edged clock per block.
2. (>2) product-term clocks per block.
3. Four Pin clocks => 3 unique clocks allowed while the fourth must be complemented.
4. Four unique pin clocks.
5. (>1) latch enable signal per block.

The following clock assignments will work:

1. One dual-edged clock per block, Two product-term clocks, One unique pin clock.
2. Two product-term clocks, Two unique pin clocks.
3. Two product term clocks, one sum-term clock.
4. One unique latch-enable signal, and all allowed clocking schemes mentioned above.

Please make note of this assignment because it will affect your ability to partition designs at the PAL block level for M5 devices.