XAPP416 (v1.0) August 7, 2002

Using an RPM Grid Macro to Control Block RAM-to-FF Timing

Author: Bret Wade

Summary

This application note describes an alternative method for specifying Relatively Placed Macros (RPMs) using a new grid system called the "RPM Grid". This grid system can be used in the Virtex™-II architectures including Virtex-II Pro™ devices. This is not a tutorial on how to create RPMs, and this document assumes some knowledge of how to create RPMs. Please see the Xilinx Libraries Guide for details on capturing RPMs. This application note will describe how to use the RPM Grid to create a heterogeneous relocatable RPM macro containing both block RAM and slice components and demonstrate how this feature can be used to optimize the timing of paths from block RAM outputs to slice registers.

Why Use an RPM Grid?

The original macro grid system does not use a universal coordinate system for all component types. This works for macros that are homogeneous or are created around the X0Y0 slice origin; however, in cases where the macro must be "normalized" (see **Normalization**), the relative position of different component types becomes corrupted.

The original macro grid system does not correspond exactly to the grid system used by the placer. Because of this, certain macro placements result in unexpected relative slice component placements. The RPM Grid solves this problem because it uses the same grid system as the placer.

Constructing an RPM Grid Macro

Relative Location RLOC constraints are applied to symbols in the logical design in the same way as a standard RPM; however, the grid values are different. The RPM Grid coordinates are determined by selecting the site in question in FPGA Editor and reading the grid coordinates in the history window. For example, selecting the lower leftmost slice site results in the following text:

```
site "SLICE_X0Y0",  type = SLICE  (RPM grid X3Y4).
```

Slice X0Y0 in the original grid system is now shown as X3Y4 in the RPM Grid system. Any symbols intended for this slice should have the following constraint applied:

```
RLOC = X3Y4
```

See **Appendix A** for an example overview of the RPM Grid coordinate system for a particular device (the Virtex-II XC2V40). FPGA Editor should generally be used to look up grid values for a specific device.

In addition to the RLOC constraints, one symbol in the macro must have the following constraint applied:

```
RPM_GRID = GRID
```

Currently, not all synthesis tools recognize and pass the RPM_GRID attribute. It may be necessary to assign this attribute using the User Constraints File (UCF) constraint:

```
INST "instance_name" RPM_GRID = GRID ;
```

where *instance_name* is the full hierarchical path to the symbol name.

Normalization

Normalization is the process by which the mapper converts the user's RLOC values to actual sites on the device in order to construct the relatively placed macro.

The components in the macros are scanned and identified for the lowest X and Y coordinates. These are then subtracted from *every* RLOC value in the macro, effectively shifting the macro into the lower-left corner of the device.

The following example uses the standard grid, given components A, B, and C:

Component	RLOC Value
A	X3Y1
B	X4Y1
C	X4Y2

The lowest X and Y is X3Y1 (Component A). Subtracting (3, 1) from every RLOC value yields:

Component	Normalized Value
A	X0Y0
B	X1Y0
C	X1Y1

The components are then placed at SLICE_X0Y0, SLICE_X1Y0, and SLICE_X1Y1.

A problem with the original grid system is that, as the various types of components (slices, block RAMs, multipliers, etc.) are on different coordinate systems, subtracting the same value from each component destroys the relative position information. For example, a user wishing to create an RPM with a slice adjacent to a block RAM might capture the RLOC values by looking in FPGA_Editor at the site names to create the following:

Component	RLOC Value
Slice	X10Y8
BRAM	X1Y1

As the SLICE_X10Y8 and RAMB16_X1Y1 sites are adjacent (on the XC2V40 device), applying the normalization rules outlined above shifts them to RLOC = X9Y7 (the slice) and RLOC = X0Y0 (the block RAM). Placing the components at SLICE_X9Y7and RAMB16_X0Y0 yields a macro in which the two sites are far apart.

When an RPM Grid is used, normalization works a bit differently. Since all of the component types are on the same coordinate system, the RLOC values can be shifted without destroying the relative placement information. However, not all coordinate values indicate a site of the correct type, or in some cases, even a legal site at all. The challenge, therefore, is to find legal placement for all of the components in the macro by scanning the grid for a match. In the case of the block RAM and slice FF example, there is a single legal placement location for each block RAM site at the lower-leftmost slice adjacent to the block RAM site.

RLOC_ORIGIN

The RLOC_ORIGIN constraint is used to assign a specific location in the device to an RPM macro. During the mapping process, the RLOC_ORIGIN constraint is converted to a MACRO LOCATE constraint in the physical constraints file (PCF). The RLOC_ORIGIN specification is a straightforward site location assignment for a standard macro that is created with the X0Y0 slice used as the lower-leftmost component. For a more complex RPM Grid macro, a user must account for the following factors:

- The RLOC_ORIGIN attribute for an RPM Grid macro must be specified in terms of RPM Grid coordinates.
- Regardless of the symbol to which the RLOC_ORIGIN attribute is assigned, the origin will be applied to the lower-leftmost component in the macro. In other words, the lower-leftmost component is always the reference component.
- If the macro requires Normalization (always the case with RPM Grid macros), the RLOC_ORIGIN coordinate should be chosen to account for this shift.

In the following example, an RPM macro is defined with the S0 slice site "SLICE_X2Y0" used for the lower-leftmost component. This corresponds to RPM Grid coordinate X6Y4. The following calculation is used to assign an RLOC_ORIGIN attribute that will place the macro in the equivalent S0 site "SLICE_X26Y40", which corresponds to RPM Grid X42Y84.

RLOC_ORIGIN = (Target site in RPM Grid) – (Macro origin in RPM Grid)

RLOC_ORIGIN = X42Y84 – X6Y4

RLOC_ORIGIN = X36Y80

Note: Currently, MAP does not accept negative RLOC_ORIGIN values. If a macro is created in the upper-right corner of a device, it will not be possible to use it in the lower-left corner. For this reason, it is good design practice is to create the macro as close to the lower-left corner as possible.

When to Use an RPM GRID

- Use the RPM Grid when constructing heterogeneous RPMs.
- Use the RPM Grid to ensure that "what you see is what you get" (WYSIWYG) when columns are spanned.

When Not to Use an RPM GRID

- Use a standard grid when constructing column-based RPMs such as carry chains. Each vertical slice increments by one.

Use Case Example: Block RAM-to-FF Timing

A good use case example for a heterogeneous RPM using the RPM Grid is a macro that combines block RAM and FFs so that the timing of the routing between the block RAM outputs and the slice FFs is optimized. In this case, the current automatic Map and Placement tools do not have enough insight into the specific device utilization required to allow 32 bits of a block RAM output bus to consistently use the optimal resources.

Consider a RAMB16_S18_S36 block RAM in an XV2V250 device using the -6 speed grade and a timing constraint of 0.7 ns for the block RAM-to-slice register routing. Without RLOC constraints, a design with 32 block RAM output-to-FF paths will fail timing on 9 of the 32 paths (See **Appendix B** for timing report). The reason for these timing failures becomes apparent when the routing resources and the mapping and placement requirements needed for all 32 paths to meet timing are examined in detail.

Optimal timing paths occur when the router is able to make use of a direct connection between the block RAM switchbox and the slice switchbox while getting through each switchbox cleanly without having to resort to a "double bounce" to reach the BX and BY pins (the slice pins used to connect directly to the FFs).

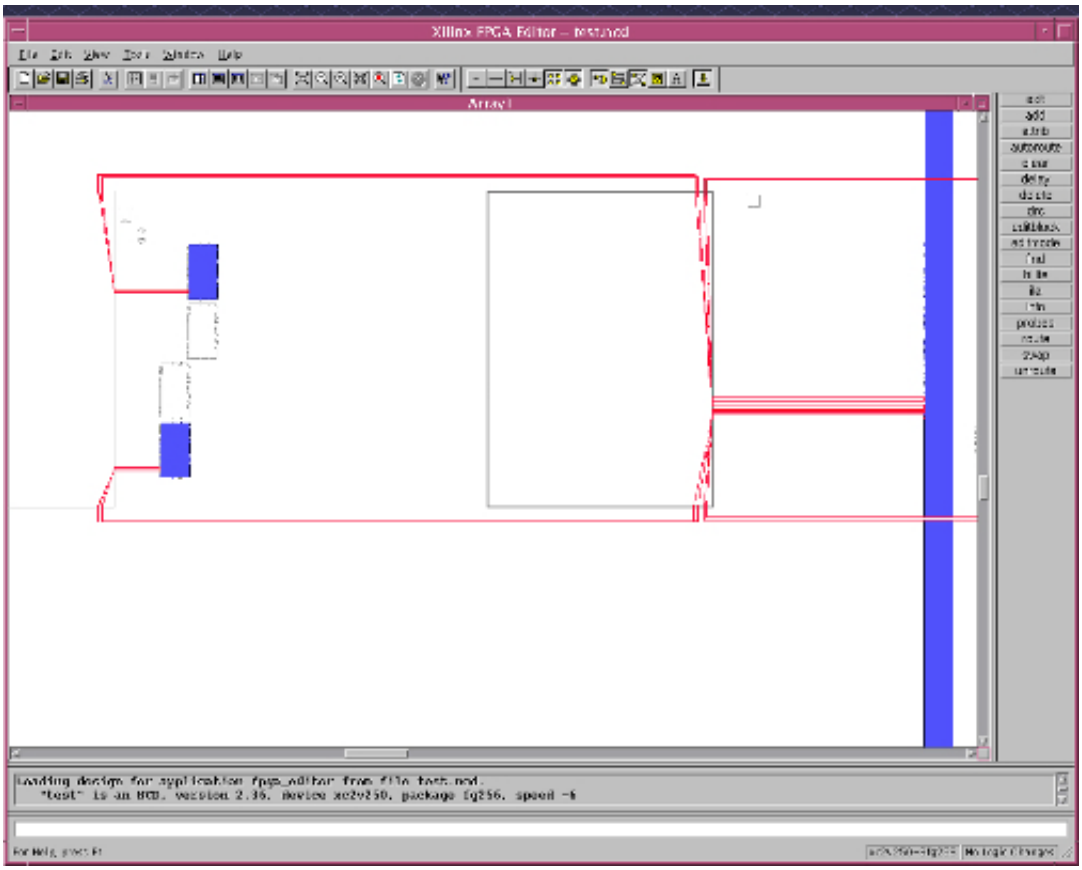


Figure 1: **Optimal Route Paths from Block RAM Outputs to Slice FFs**

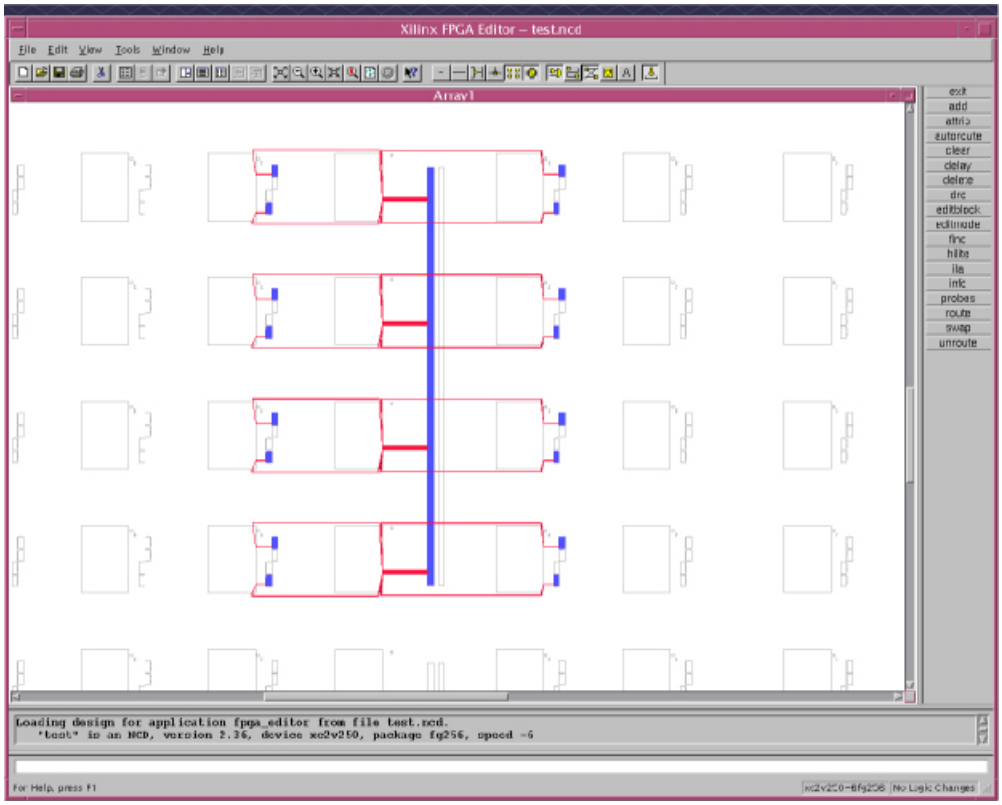
After analyzing how such a connection can be allocated for all 32 bits of the block RAM output bus, the following three conclusions can be made:

1. Because there are dedicated switchbox paths to both BX and BY pins, it is important to balance the register utilization evenly between the FFX and FFY slice Basic Elements of Logic (BELs).
2. As there are switchbox-to-switchbox direct connections in both the left and right directions, it is important to balance the FF RLOCs evenly to columns on both the left and right sides of the block RAM.
3. As the block RAM outputs are broken up into groups of eight pins per CLB row, the FF RLOCs should also be partitioned into groups of eight per CLB row.

The necessary RLOC constraints must result in the following utilization:

1. Two flip-flops per slice to balance BX and BY pin usage;
2. Two slices per CLB (top and bottom slices chosen arbitrarily);
3. Two CLBs per row, using columns on either side of the block RAM to balance the switchbox direct connection usage.

The automatic map and placement tools do not have enough insight into the routing resources needed to precisely balance the resource usage; however, the use of RLOC constraints will ensure this balance. See **Appendix C** for an example of RLOCs that successfully distribute the FFs around the block RAM and **Appendix D** for the resulting timing report.



x416_02_062602

Figure 2: **Example of Successful Routing of 32-bit Block RAM Output Bus**

To examine this example design, including the VHDL source, plesae see the 4.1i ISE project at **ftp://ftp.xilinx.com/pub/applications/xapp/xapp416.zip**.

The RPM Grid macro in this example is compatible with both the Virtex-II and Virtex-II Pro architectures. The example design can be retargeted to a Virtex-II Pro device by changing the target device and specifying an appropriate RLOC_ORIGIN value (or none) for the new device in the UCF file.

Conclusion

These results illustrate how a heterogeneous RPM using the RPM Grid can be successfully used to ensure that specific hardware utilization is obtained. In turn, this allows the router to make use of the optimal routing resource for this specific implementation.

Appendix A The RPM Grid

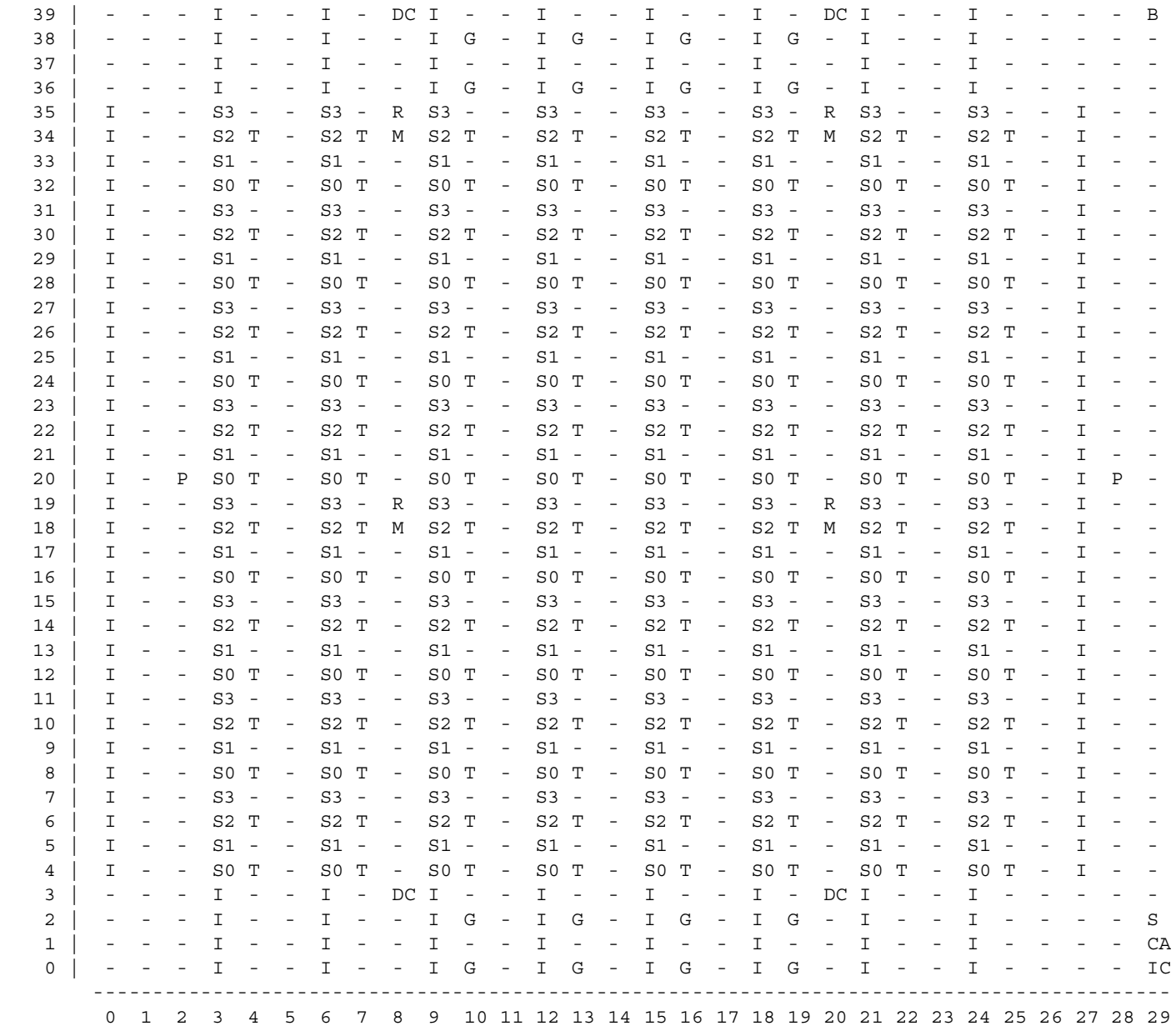


Figure 3: **Virtex-II XC2V40 Device Layout**

Table 1: **Layout Grid Key**

Symbol	Component
S0, S1, S2, S3	Slices S0, S1, S2 and S3
CA	CAPTURE
DC	DCM
G	GCLK
I	IOB
IC	ICAP
B	BSCAN
M	Multiplier
P	PCILO
R	Block RAM
S	Startup
T	TBUF

Appendix B Timing Results of Non-RLOC'd Design

```
------------------------------------------------------------------------------
  Constraint                                | Requested  | Actual     | Logic
                                            |            |            | Levels
------------------------------------------------------------------------------
  NET "dob_0" MAXDELAY = 700 pS             | 0.700ns    | 0.685ns    |
------------------------------------------------------------------------------
  NET "dob_1" MAXDELAY = 700 pS             | 0.700ns    | 0.672ns    |
------------------------------------------------------------------------------
  NET "dob_10" MAXDELAY = 700 pS            | 0.700ns    | 0.685ns    |
------------------------------------------------------------------------------
  NET "dob_11" MAXDELAY = 700 pS            | 0.700ns    | 0.672ns    |
------------------------------------------------------------------------------
  NET "dob_12" MAXDELAY = 700 pS            | 0.700ns    | 0.681ns    |
------------------------------------------------------------------------------
  NET "dob_13" MAXDELAY = 700 pS            | 0.700ns    | 0.668ns    |
------------------------------------------------------------------------------
  NET "dob_14" MAXDELAY = 700 pS            | 0.700ns    | 0.688ns    |
------------------------------------------------------------------------------
* NET "dob_15" MAXDELAY = 700 pS            | 0.700ns    | 0.884ns    |
------------------------------------------------------------------------------
* NET "dob_16" MAXDELAY = 700 pS            | 0.700ns    | 0.920ns    |
------------------------------------------------------------------------------
* NET "dob_17" MAXDELAY = 700 pS            | 0.700ns    | 0.879ns    |
------------------------------------------------------------------------------
* NET "dob_18" MAXDELAY = 700 pS            | 0.700ns    | 0.915ns    |
------------------------------------------------------------------------------
* NET "dob_19" MAXDELAY = 700 pS            | 0.700ns    | 1.112ns    |
------------------------------------------------------------------------------
  NET "dob_2" MAXDELAY = 700 pS             | 0.700ns    | 0.685ns    |
------------------------------------------------------------------------------
  NET "dob_20" MAXDELAY = 700 pS            | 0.700ns    | 0.659ns    |
------------------------------------------------------------------------------
  NET "dob_21" MAXDELAY = 700 pS            | 0.700ns    | 0.668ns    |
------------------------------------------------------------------------------
  NET "dob_22" MAXDELAY = 700 pS            | 0.700ns    | 0.686ns    |
```

```
------------------------------------------------------------------------------
  NET "dob_23" MAXDELAY = 700 pS            | 0.700ns     | 0.646ns     |
------------------------------------------------------------------------------
  NET "dob_24" MAXDELAY = 700 pS            | 0.700ns     | 0.686ns     |
------------------------------------------------------------------------------
  NET "dob_25" MAXDELAY = 700 pS            | 0.700ns     | 0.673ns     |
------------------------------------------------------------------------------
  NET "dob_26" MAXDELAY = 700 pS            | 0.700ns     | 0.681ns     |
------------------------------------------------------------------------------
  NET "dob_27" MAXDELAY = 700 pS            | 0.700ns     | 0.668ns     |
------------------------------------------------------------------------------
* NET "dob_28" MAXDELAY = 700 pS            | 0.700ns     | 0.902ns     |
------------------------------------------------------------------------------
* NET "dob_29" MAXDELAY = 700 pS            | 0.700ns     | 1.150ns     |
------------------------------------------------------------------------------
  NET "dob_3" MAXDELAY = 700 pS             | 0.700ns     | 0.672ns     |
------------------------------------------------------------------------------
* NET "dob_30" MAXDELAY = 700 pS            | 0.700ns     | 1.260ns     |
------------------------------------------------------------------------------
* NET "dob_31" MAXDELAY = 700 pS            | 0.700ns     | 1.261ns     |
------------------------------------------------------------------------------
  NET "dob_4" MAXDELAY = 700 pS             | 0.700ns     | 0.685ns     |
------------------------------------------------------------------------------
  NET "dob_5" MAXDELAY = 700 pS             | 0.700ns     | 0.668ns     |
------------------------------------------------------------------------------
  NET "dob_6" MAXDELAY = 700 pS             | 0.700ns     | 0.685ns     |
------------------------------------------------------------------------------
  NET "dob_7" MAXDELAY = 700 pS             | 0.700ns     | 0.672ns     |
------------------------------------------------------------------------------
  NET "dob_8" MAXDELAY = 700 pS             | 0.700ns     | 0.659ns     |
------------------------------------------------------------------------------
  NET "dob_9" MAXDELAY = 700 pS             | 0.700ns     | 0.668ns     |
------------------------------------------------------------------------------
```

Appendix C

Constraints Used

```
RLOC constraints from VHDL file:
attribute RLOC: string;
attribute RLOC of DOB0: label is "x6y4";
attribute RLOC of DOB1: label is "x6y4";
attribute RLOC of DOB2: label is "x9y4";
attribute RLOC of DOB3: label is "x9y4";
attribute RLOC of DOB4: label is "x6y8";
attribute RLOC of DOB5: label is "x6y8";
attribute RLOC of DOB6: label is "x9y8";
attribute RLOC of DOB7: label is "x9y8";
attribute RLOC of DOB8: label is "x6y12";
attribute RLOC of DOB9: label is "x6y12";
attribute RLOC of DOB10: label is "x9y12";
attribute RLOC of DOB11: label is "x9y12";
attribute RLOC of DOB12: label is "x6y16";
attribute RLOC of DOB13: label is "x6y16";
attribute RLOC of DOB14: label is "x9y16";
attribute RLOC of DOB15: label is "x9y16";
attribute RLOC of DOB16: label is "x6y7";
attribute RLOC of DOB17: label is "x6y7";
attribute RLOC of DOB18: label is "x9y7";
attribute RLOC of DOB19: label is "x9y7";
attribute RLOC of DOB20: label is "x6y11";
attribute RLOC of DOB21: label is "x6y11";
```

```
attribute RLOC of DOB22: label is "x9y11";
attribute RLOC of DOB23: label is "x9y11";
attribute RLOC of DOB24: label is "x6y15";
attribute RLOC of DOB25: label is "x6y15";
attribute RLOC of DOB26: label is "x9y15";
attribute RLOC of DOB27: label is "x9y15";
attribute RLOC of DOB28: label is "x6y19";
attribute RLOC of DOB29: label is "x6y19";
attribute RLOC of DOB30: label is "x9y19";
attribute RLOC of DOB31: label is "x9y19";
attribute RLOC of u10: label is "x8y19";

attribute U_SET: string;
attribute U_SET of DOB0: label is "blkram_ff";
attribute U_SET of DOB1: label is "blkram_ff";
attribute U_SET of DOB2: label is "blkram_ff";
attribute U_SET of DOB3: label is "blkram_ff";
attribute U_SET of DOB4: label is "blkram_ff";
attribute U_SET of DOB5: label is "blkram_ff";
attribute U_SET of DOB6: label is "blkram_ff";
attribute U_SET of DOB7: label is "blkram_ff";
attribute U_SET of DOB8: label is "blkram_ff";
attribute U_SET of DOB9: label is "blkram_ff";
attribute U_SET of DOB10: label is "blkram_ff";
attribute U_SET of DOB11: label is "blkram_ff";
attribute U_SET of DOB12: label is "blkram_ff";
attribute U_SET of DOB13: label is "blkram_ff";
attribute U_SET of DOB14: label is "blkram_ff";
attribute U_SET of DOB15: label is "blkram_ff";
attribute U_SET of DOB16: label is "blkram_ff";
attribute U_SET of DOB17: label is "blkram_ff";
attribute U_SET of DOB18: label is "blkram_ff";
attribute U_SET of DOB19: label is "blkram_ff";
attribute U_SET of DOB20: label is "blkram_ff";
attribute U_SET of DOB21: label is "blkram_ff";
attribute U_SET of DOB22: label is "blkram_ff";
attribute U_SET of DOB23: label is "blkram_ff";
attribute U_SET of DOB24: label is "blkram_ff";
attribute U_SET of DOB25: label is "blkram_ff";
attribute U_SET of DOB26: label is "blkram_ff";
attribute U_SET of DOB27: label is "blkram_ff";
attribute U_SET of DOB28: label is "blkram_ff";
attribute U_SET of DOB29: label is "blkram_ff";
attribute U_SET of DOB30: label is "blkram_ff";
attribute U_SET of DOB31: label is "blkram_ff";
attribute U_SET of u10: label is "blkram_ff";

UCF constraints used:
NET "dob*" MAXDELAY = .7 ns ;
INST "DOB0" RLOC_ORIGIN = X36Y80 ;
INST "DOB0" RPM_GRID = GRID;
```

```
21 |  I  -  -  S1  -  -  S1  -  -  S1  -  -  S1  -  -  S1  -  -  S1  -  -
20 |  I  -  P  S0  T  -  S0  T  -  S0  T  -  S0  T  -  S0  T  -  S0  T  -
19 |  I  -  -  S3  -  -  S3  -  R  S3  -  -  S3  -  -  S3  -  -  S3  -  R
18 |  I  -  -  S2  T  -  S2  T  M  S2  T  -  S2  T  -  S2  T  -  S2  T  M
17 |  I  -  -  S1  -  -  S1  -  -  S1  -  -  S1  -  -  S1  -  -  S1  -  -
16 |  I  -  -  S0  T  -  S0  T  -  S0  T  -  S0  T  -  S0  T  -  S0  T  -
15 |  I  -  -  S3  -  -  S3  -  -  S3  -  -  S3  -  -  S3  -  -  S3  -  -
14 |  I  -  -  S2  T  -  S2  T  -  S2  T  -  S2  T  -  S2  T  -  S2  T  -
13 |  I  -  -  S1  -  -  S1  -  -  S1  -  -  S1  -  -  S1  -  -  S1  -  -
12 |  I  -  -  S0  T  -  S0  T  -  S0  T  -  S0  T  -  S0  T  -  S0  T  -
11 |  I  -  -  S3  -  -  S3  -  -  S3  -  -  S3  -  -  S3  -  -  S3  -  -
10 |  I  -  -  S2  T  -  S2  T  -  S2  T  -  S2  T  -  S2  T  -  S2  T  -
 9 |  I  -  -  S1  -  -  S1  -  -  S1  -  -  S1  -  -  S1  -  -  S1  -  -
 8 |  I  -  -  S0  T  -  S0  T  -  S0  T  -  S0  T  -  S0  T  -  S0  T  -
 7 |  I  -  -  S3  -  -  S3  -  -  S3  -  -  S3  -  -  S3  -  -  S3  -  -
 6 |  I  -  -  S2  T  -  S2  T  -  S2  T  -  S2  T  -  S2  T  -  S2  T  -
 5 |  I  -  -  S1  -  -  S1  -  -  S1  -  -  S1  -  -  S1  -  -  S1  -  -
 4 |  I  -  -  S0  T  -  S0  T  -  S0  T  -  S0  T  -  S0  T  -  S0  T  -
 3 |  -  -  -  I   -  -  I   -  DC I   -  -  I   -  -  I   -  -  I   -  DC
 2 |  -  -  -  I   -  -  I   -  -  I   G  -  I   G  -  I   G  -  I   G  -
 1 |  -  -  -  I   -  -  I   -  -  I   -  -  I   -  -  I   -  -  I   -  -
 0 |  -  -  -  I   -  -  I   -  -  I   G  -  I   G  -  I   G  -  I   G  -
   ------------------------------------------------------------------------
    0  1  2  3  4  5  6  7  8  9  10 11 12 13 14 15 16 17 18 19 20
```

Figure 4: **Component Site Utilization of Macro Indicated by Bold Text**

Appendix D Timing Results of RLOC'D Design

Constraint	Requested	Actual	Logic Levels
NET "dob_0" MAXDELAY = 700 pS	0.700ns	0.659ns	
NET "dob_1" MAXDELAY = 700 pS	0.700ns	0.668ns	
NET "dob_10" MAXDELAY = 700 pS	0.700ns	0.686ns	
NET "dob_11" MAXDELAY = 700 pS	0.700ns	0.646ns	
NET "dob_12" MAXDELAY = 700 pS	0.700ns	0.659ns	
NET "dob_13" MAXDELAY = 700 pS	0.700ns	0.668ns	
NET "dob_14" MAXDELAY = 700 pS	0.700ns	0.686ns	
NET "dob_15" MAXDELAY = 700 pS	0.700ns	0.646ns	
NET "dob_16" MAXDELAY = 700 pS	0.700ns	0.659ns	
NET "dob_17" MAXDELAY = 700 pS	0.700ns	0.668ns	
NET "dob_18" MAXDELAY = 700 pS	0.700ns	0.686ns	
NET "dob_19" MAXDELAY = 700 pS	0.700ns	0.646ns	
NET "dob_2" MAXDELAY = 700 pS	0.700ns	0.686ns	
NET "dob_20" MAXDELAY = 700 pS	0.700ns	0.659ns	

```
------------------------------------------------------------------------------
  NET "dob_21" MAXDELAY = 700 pS            | 0.700ns    | 0.668ns    |
------------------------------------------------------------------------------
  NET "dob_22" MAXDELAY = 700 pS            | 0.700ns    | 0.686ns    |
------------------------------------------------------------------------------
  NET "dob_23" MAXDELAY = 700 pS            | 0.700ns    | 0.646ns    |
------------------------------------------------------------------------------
  NET "dob_24" MAXDELAY = 700 pS            | 0.700ns    | 0.659ns    |
------------------------------------------------------------------------------
  NET "dob_25" MAXDELAY = 700 pS            | 0.700ns    | 0.668ns    |
------------------------------------------------------------------------------
  NET "dob_26" MAXDELAY = 700 pS            | 0.700ns    | 0.686ns    |
------------------------------------------------------------------------------
  NET "dob_27" MAXDELAY = 700 pS            | 0.700ns    | 0.646ns    |
------------------------------------------------------------------------------
  NET "dob_28" MAXDELAY = 700 pS            | 0.700ns    | 0.659ns    |
------------------------------------------------------------------------------
  NET "dob_29" MAXDELAY = 700 pS            | 0.700ns    | 0.669ns    |
------------------------------------------------------------------------------
  NET "dob_3" MAXDELAY = 700 pS             | 0.700ns    | 0.646ns    |
------------------------------------------------------------------------------
  NET "dob_30" MAXDELAY = 700 pS            | 0.700ns    | 0.687ns    |
------------------------------------------------------------------------------
  NET "dob_31" MAXDELAY = 700 pS            | 0.700ns    | 0.646ns    |
------------------------------------------------------------------------------
  NET "dob_4" MAXDELAY = 700 pS             | 0.700ns    | 0.659ns    |
------------------------------------------------------------------------------
  NET "dob_5" MAXDELAY = 700 pS             | 0.700ns    | 0.668ns    |
------------------------------------------------------------------------------
  NET "dob_6" MAXDELAY = 700 pS             | 0.700ns    | 0.686ns    |
------------------------------------------------------------------------------
  NET "dob_7" MAXDELAY = 700 pS             | 0.700ns    | 0.646ns    |
------------------------------------------------------------------------------
  NET "dob_8" MAXDELAY = 700 pS             | 0.700ns    | 0.659ns    |
------------------------------------------------------------------------------
  NET "dob_9" MAXDELAY = 700 pS             | 0.700ns    | 0.668ns    |
------------------------------------------------------------------------------
```

Revision History

The following table shows the revision history for this document.

Date	Version	Revision
08/07/02	1.0	Initial Xilinx release.