

---

# **QuickLogic-FPGA-Toolchain**

***Release 1.0.0***

**Lalit Narain Sharma**

**Jul 26, 2021**



# CONTENTS

<b>1</b>	<b>Symbiflow Installation Guide and Tutorial</b>	<b>1</b>
1.1	Installing Symbiflow on Linux . . . . .	1
<b>2</b>	<b>QLF-K4N8/QLF-K6N10 Device</b>	<b>3</b>
2.1	Symbiflow: Design flow (QLF-K4N8/QLF-K6N10) . . . . .	3
2.2	Supported Commands . . . . .	5
2.3	Design flow . . . . .	6
2.3.1	Performing the Pre-Layout Simulation . . . . .	6
2.3.2	Performing Design Synthesis . . . . .	7
2.3.3	Pin Mapping . . . . .	8
2.3.4	Running Pack, Place and Route Tools . . . . .	16
2.3.5	Performing the Post-Layout Timing Simulation . . . . .	17
2.3.6	Generate the Programming Files . . . . .	17
2.3.7	PCF Sample . . . . .	18
2.4	BRAM and DSP . . . . .	38
2.5	Online References . . . . .	40
<b>3</b>	<b>S3B Device</b>	<b>41</b>
3.1	Symbiflow: Design flow (EOS-S3) . . . . .	41
3.1.1	Supported Commands . . . . .	43
3.2	Run design flow on a simple counter design (EOS-S3) . . . . .	43
3.2.1	Performing the Pre-Layout Simulation . . . . .	44
3.2.2	Performing Design Synthesis . . . . .	45
3.2.3	Running pack, Place and Route tools (EOS-S3) . . . . .	45
3.2.4	Performing the Post-Layout Simulation (verifying the configuration bits) . . . . .	46
3.2.5	Performing the Post-Layout Timing Simulation . . . . .	47
3.2.6	Generate the Jlink and openOCD file . . . . .	47
3.2.7	Generate the ASCII header file format . . . . .	48
3.2.8	Generate the Binary File format . . . . .	48
3.2.9	PCF Sample . . . . .	53
3.3	Hardware Limitations and Online References . . . . .	55
3.4	S3B Device . . . . .	55
3.4.1	RAM Features . . . . .	55
3.4.2	FIFO Features . . . . .	55
3.4.3	Multiplier Features . . . . .	56
3.4.4	Macro Usage and examples . . . . .	56
	<b>Index</b>	<b>65</b>



## SYMBIFLOW INSTALLATION GUIDE AND TUTORIAL

This provides the details of the Symbiflow package installation and the various commands supported by the tool. It covers how to install Symbiflow on the Linux operating systems and the usage of the tool by going over a simple example.

### System Requirements

Require-ments	Linux	CentOS	Ubuntu
Processor	Intel Xeon® or similar proces-sors	Intel Xeon or similar proces-sors	Intel Xeon or similar proces-sors

Ram Size: 2 GB or more Free Hard-Disc space: 5GB or more

## 1.1 Installing Symbiflow on Linux

Download the required symbiflow installer from the Releases tag.

To install Symbiflow on Linux:

1. Set the execute permission for the .run file

```
chmod 755 Symbiflow*<version>*.gz.run
```

2. Set the <INSTALL\_DIR>: variable:

```
export INSTALL_DIR=<Install path>
```

3. Execute the .run file from the terminal:

```
bash Symbiflow<*version>*.gz.run
```

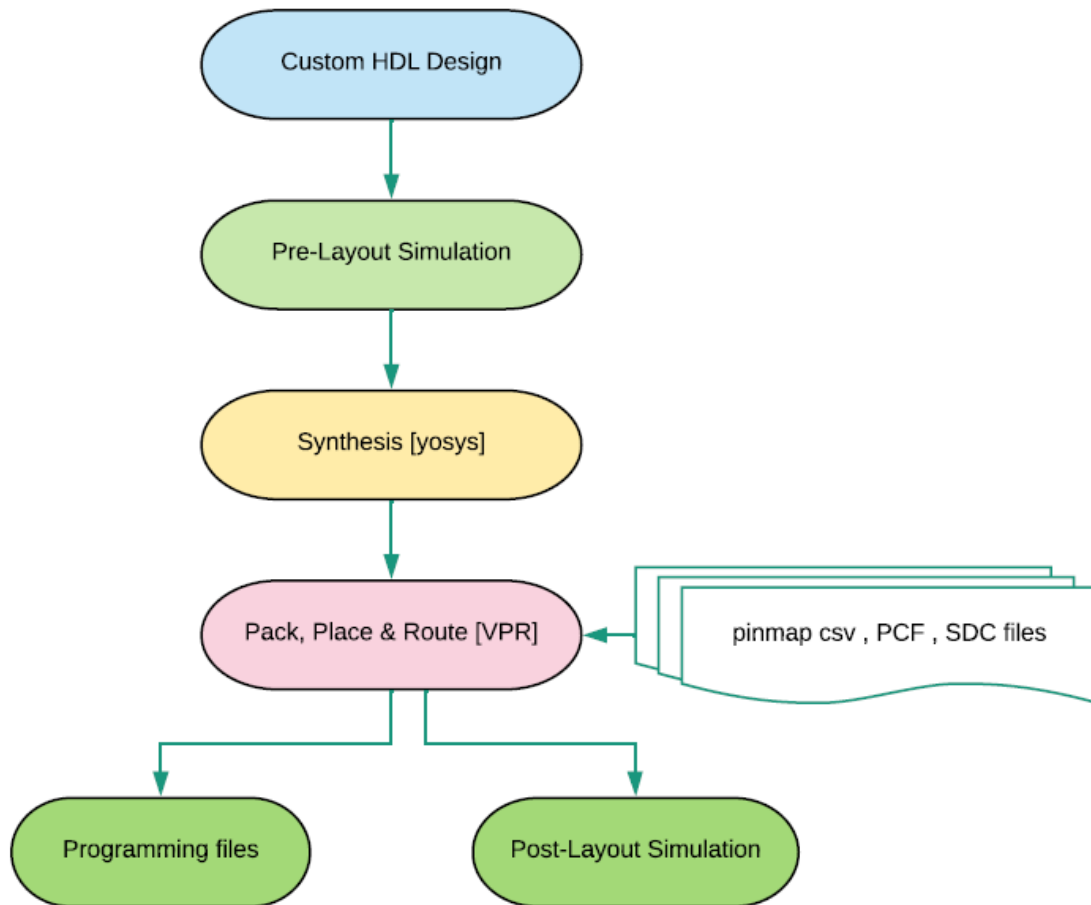


## QLF-K4N8/QLF-K6N10 DEVICE

This provides details about running designs using qlf-k4n8/qlf-k6n10 device on Symbiflow

### 2.1 Symbiflow: Design flow (QLF-K4N8/QLF-K6N10)

Symbiflow design flow starts with the verilog as input, synthesis done using the yosys, pack, place and route with VPR tool. Finally the programming files are dumped which can be used for the fpga programming. Post layout verilog files can be used to validate the verilog design netlist dumped after the place and route.





## 2.2 Supported Commands

Command option	Represented for (One with * supports only for qlf_k4n8)	Options
<b>-synth</b>	Synthesis using yosys	-
<b>-compile</b>	Run pack, place, route and generate fasm file	-
<b>-src</b> <b>&lt;source path&gt;</b>	Source file folder	-
<b>-d</b> <b>&lt;device&gt;</b>	Device supported	qlf_k4n8/qlf_k6n10
<b>-P</b> <b>&lt;package&gt;</b>	*Package csv file	qlf_k4n8 csv file
<b>-p</b> <b>&lt;pcf file&gt;</b>	*Fix Placement constraints of IO's	-
<b>-s</b> <b>&lt;sdc file&gt;</b>	*Timing Constraint File (SDC)	Refer online documents section for SDC constraints supported
<b>-r</b> <b>&lt;router flag&gt;</b>	Timing: means no attention is paid to delay. Congestion: means nets on the critical path pay no attention to congestion	timing, congestion
<b>-t</b> <b>&lt;top module&gt;</b>	Top module of the Verilog design	-
<b>-v</b> <b>&lt;Verilog list files&gt;</b>	Verilog source files	Only Verilog supported
<b>-pnr_corner</b>	*Timing corner for the place and route tool	fast/slow
<b>-analysis_corner</b>	*SDF corner for the verilog post layout	fast/slow
<b>-dump</b>	Dump post layout verilog file	post_verilog

Below commands are supported only for synthesis(-synth option):

Command option	Represented for	Usage
<b>-y</b>	Specifies a verilog library directory to search for module definition. Multiple directories can be specified with space	<b>-y</b> <b>&lt;directory_path&gt;</b>
<b>-f</b>	Specifies a file that contains a list of commands to run	<b>-f</b> <b>&lt;filename&gt;</b>
<b>+in-cdir+</b>	Specifies the directories that contains the files declared with the <code>`include</code> compiler directive. Multiple directories can be specified with the + character	<b>+in-cdir+</b> <b>&lt;directory&gt;</b>
<b>+libext+</b>	Specify files with particular extension in the directory. You can specify more than one extension, separating each extension with the + character. Use this option when you enter the -y option	<b>+libext+</b> <b>&lt;extension&gt;</b>
<b>+define+</b>	Defines a text macro. Use <code>`ifdef</code> compiler directive in your Verilog source code	<b>+define+</b> <b>&lt;macro_name&gt;=&lt;value&gt;</b>

## 2.3 Design flow

### Setup environment

To run the counter\_16bit or any example, perform these steps once. Design flow will be covered using the counter\_16bit design which is available in the package. We will use the device QLF\_K4N8 for this run. However the flow for the device QLF\_K6N10 is similar except that it is supported only till place and route.

```
export INSTALL_DIR="specify the installpath"
cd $INSTALL_DIR; source setup.sh

OR

#adding symbiflow toolchain binaries to PATH
export PATH="$INSTALL_DIR/quicklogic-arch-defs/bin:$INSTALL_DIR/quicklogic-
↪arch-defs/bin/python:$PATH"
source "$INSTALL_DIR/conda/etc/profile.d/conda.sh"
conda activate
```

Entering an HDL Design:

1. Write a Verilog code for the design using any text editor.
2. Verify the syntax.
3. Create the simulation stimuli using any text editor.

The code and testbench for the example design are present at: `<Install_Path>/quicklogic-arch-defs/share/symbiflow/tests/counter_16bit/`

### 2.3.1 Performing the Pre-Layout Simulation

To perform a pre-layout simulation:

#### Using Icarus Verilog:

To create the VCD output file that will be used to perform graphical analysis of the Design, the following lines are added in the TB:

```
initial begin
    $dumpfile("counter_16bit_tb.vcd");
    $dumpvars(0, counter_16bit_tb);
    $display("\t\ttime,\t\tclk,\t\treset,\t\ttenable,\t\tcount");
    $monitor("%d,\t\tb,\t\tb,\t\tb,\t\tb", $time, clk, reset, enable,
↪count);
end
```

The “iverilog” and “vvp” commands are the most important commands available to users of Icarus Verilog. The “iverilog” command is the compiler, and the “vvp” command is the simulation runtime engine.

```
cd <INSTALL_PATH>/quicklogic-arch-defs/share/symbiflow/tests/
↪counter_16bit
```

The “iverilog” command supports multi-file designs by two methods. The simplest is to list the files on the command line:

```
iverilog -o my_design counter_16bit.v counter_16bit_tb.v
vvp my_design
```

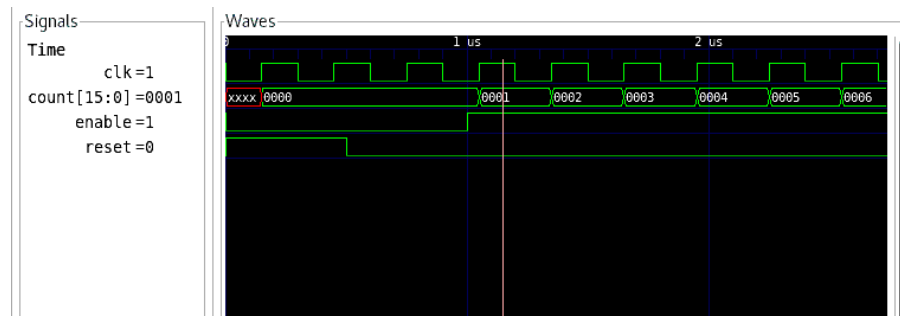
This command compiles the design, which is spread across two input files, and generates the compiled result into the “my\_design” file. Another technique is to use a commandfile, which lists the input files in a text file. For example, create a text file called “file\_list.txt” with the files listed one per line:

```
#Inside file_list.txt
counter_16bit.v
counter_16bit_tb.v
```

```
#Then compile and execute the design with command:
iverilog -o my_design -c file_list.txt
vvp my_design
```

VCD file is created, it can be viewed using GTKWave:

```
gtkwave counter_16bit_tb.vcd &
```



## 2.3.2 Performing Design Synthesis

To perform a design synthesis:

In SymbiFlow, the synthesis of Verilog files is performed with Yosys. Yosys parses Verilog files, applies basic optimizations, performs technological mapping to FPGA blocks, and generates JSON and EBLIF files for the place and route tool. Example is for qlf\_k4n8.

Syntax:

```
ql_symbiflow -synth -src <source complete path> -d <device> -t <top_
module name> -v <verilog files>
```

```
cd <INSTALL_PATH>/quicklogic-arch-defs/share/symbiflow/tests/counter_
16bit
```

and run the below command:

```
ql_symbiflow -synth -d qlf_k4n8 -t top -v counter_16bit.v
```

Output files for synthesis are present in ‘build’ folder:

<TOP>.eblif : netlist file for the design <TOP>\_synth.log : synthesis log information, refer this file for any issues during synthesis

Resource utilization in the top\_synth.log of the counter:

Number of wires:	23
Number of wire bits:	68
Number of public wires:	23
Number of public wire bits:	68
Number of memories:	0
Number of memory bits:	0
Number of processes:	0
Number of cells:	48
dff	16
lut	16
adder_lut4	16

---

**Note:** > All the output log files will be dumped in {source path}/build folder > -src command is optional if run from the same directory where source files are present. > -synth option is supported for both devices **QLF\_K4N8 & QLF\_K6N10**.

---

### 2.3.3 Pin Mapping

#### Table of Contents

- *Pin Mapping*
  - *Scope*
  - *XML File Specification*
  - *CSV File Specification*
  - *SDC File Specification*

Document for pin mapping file specification

This describes in detail about the xml and csv file format that are used to define the pin mapping.

#### Scope

This requirement document lists down the requirements for a pin mapping specification file. This file format addresses the requirements for specifying pin mapping of an eFPGA device interface ports to end-user defined package (explained in below example)

## Example

As an example, refer to [OpenFPGA Caravel architecture](#)

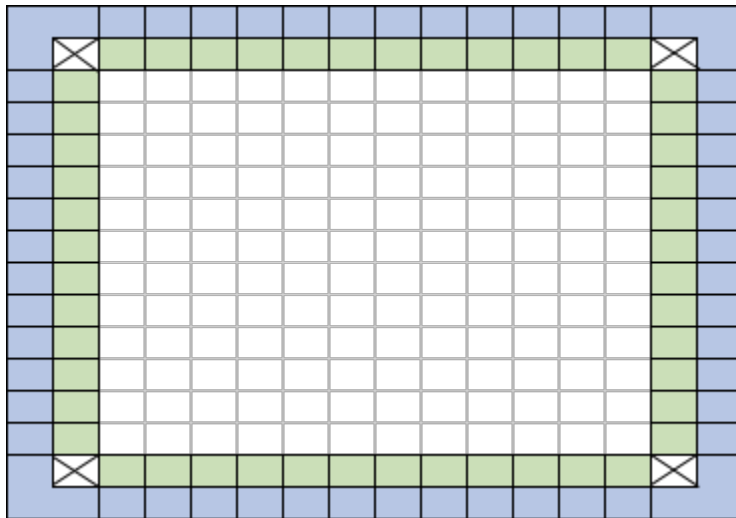
Its I/O resources are explained [here](#)

To summarize, it is 12x12 size eFPGA architecture with 144 I/Os that consists of:

- **29 external I/Os** are accessible through the Caravel SoC's \_General-Purpose I/Os (GPIOs).
- **115 internal I/Os** are accessible through the Caravel SOC's logic analyzer and wishbone interfaces, which are controlled by the RISC-V processor.

Note that in eFPGA device, each IO interface can have multiple IO ports (like 16 input/16 output at each location). However, for the current architecture, only 144 I/Os are required. For some other architecture, the same eFPGA can be used with different I/O setup.

We require a mechanism to specify pin-mapping for GPIOs with user-defined pin-names rather than default names present in eFPGA I/O interface.



Above figure is a 12x12 eFPGA device where green-colored cells represent eFPGA IO interface cells and blue-colored cells represent user-defined package:

- Green colored cells are eFPGA interface cells. Each cell consists of 16 input/16 output ports
- White colored cells are eFPGA core of 12x12 device size
- Blue colored cells are termed as an end-user package. Here end-user is an eFPGA device user who intends to use it in their silicon. \* User defines their pin-mapping to green-colored IO interface ports \* User can define multiple such packages

In this example, we require the following pin-mapping specifications:

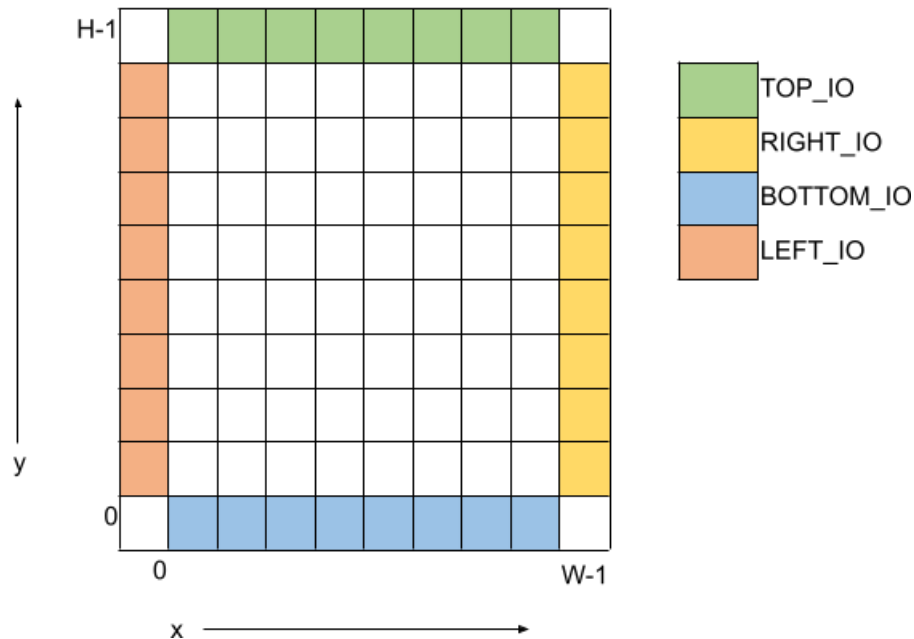
1. **Internally (not for the eFPGA user), we need to define a mapping between IO cell ports (defined in vpr\_arch xml file) to E**
  - a. This specification is defined in XML file (details are mentioned in the below section)
2. **For eFPGA users (SoC vendors using the eFPGA fabric in their IP), define pin-mapping between IO interface cell ports (**
  - a. This is defined in a comma-separated (csv) file, details are mentioned in the following section.
  - b. User can associate the defined pins with clock as well as can define timing constraints in a sdc file

## XML File Specification

XML file specification is primarily to define the mapping of the interface cell ports defined in vpr\_arch xml, to the EFPGA IO interface port names. This mapping is required by Symbiflow alongwith architecture definition file i.e. vpr\_arch xml file. Symbiflow will process this file and use this information for IO placement and then later on use this to map it with the user-defined pin-mapping file.

As shown in the below diagram that represents 8x8 device, it follows the similar convention as represented in vpr arch xml.

- Base coordinates start with the bottom-left corner.
- Width (W) represents the number of columns.
- Height (H) represents the number of rows.
- x coordinate can be an integer between 0 to W-1.
- y coordinate can be an integer between 0 to H-1.
- z coordinate can be an integer to represent the number of items stacked at a specific x,y location.



The xml file follows these specifications:

1. In XML file, sections and subsections are defined in upper caps.
2. **Root section in the xml is named as “DEVICE” and it has the following attributes defined:**
  - a. “name”: Define the device architecture name. **Mandatory attribute.**
  - b. “family”: Define the device architecture family name. **Mandatory attribute**
  - c. “width”: Define the width or the number of cells in a row. Use “W” as a special variable to represent the width of the device in xml specification. **Mandatory attribute.**
  - d. “height”: Define the height or the number of cells in a column. Use “H” as a special variable to represent the height of the device in xml specification. **Mandatory attribute.**
  - e. “z”: Define number of ports stacked at a specific row/column. By default, z is “1”.

3. Root section has a child section by name – “IO”.
4. **“IO” has the following child sections:**
  - a. **“TOP\_IO”: It defines the interface cells & ports at the top-side of the device. It has the following attributes defined:**
    1. “y”: Defines the row number at which top-level interface cells are present. By default y for TOP\_IO is “H-1”.
  - b. **“RIGHT\_IO”: It defines the interface cells & ports at the right-side of the device. It has the following attributes defined:**
    2. “x”: Defines the column number at which right-level interface cells are present. By default x for RIGHT\_IO is “W-1”.
  - c. **“BOTTOM\_IO”: It defines the interface cells & ports at the bottom-level of the device. It has the following attributes defined:**
    3. “y”: Defines the row number at which bottom-level interface cells are present. By default y for BOTTOM\_IO is “0”.
  - d. **“LEFT\_IO”: It defines the interface cells & ports at the left-side of the device. It has the following attributes defined:**
    4. “x”: Defines the column number at which left-level interface cells are present. By default x for LEFT\_IO is “0”.
5. Sub-section “TOP\_IO”, “BOTTOM\_IO”, “LEFT\_IO” or “RIGHT\_IO” section has the following sub-section defined:
  - a. **“CELL”: Defines the cell level port mapping. It accepts the following attributes:**
    - i. “port\_name”: Port name (can be scalar, bus or part-select) defined at interface cell-level. Corresponds to port name as defined in IO cell port in vpr\_arch xml model section. **Mandatory attribute**
    - ii. “mapped\_name”: Mapped IO interface port name. In case of output port, the default value is “NA” and for input port, the default value is “GND”
    - iii. “startx” or “starty”: For TOP\_IO & BOTTOM\_IO, “startx” is defined as the starting column number in the given TOP\_IO or BOTTOM\_IO section, where the given pin mapping starts for bus ports. “starty” is defined as the starting row number in the given LEFT\_IO or RIGHT\_IO section, where the given pin mapping starts for bus ports. NOTE: it is an error if “startx” is defined in the LEFT\_IO or RIGHT\_IO section. Similarly, it is an error if “starty” is defined in the TOP\_IO & BOTTOM\_IO section.
    - iv. “endx” or “endy”: For TOP\_IO & BOTTOM\_IO, “endx” is defined as the ending column number in the given TOP\_IO or BOTTOM\_IO section, where the given pin mapping ends for bus ports. “endy” is defined as the ending row number in the given LEFT\_IO or RIGHT\_IO section, where the given pin mapping ends for bus ports. NOTE: it is an error if “endx” is defined in the LEFT\_IO or RIGHT\_IO section. Similarly, it is an error if “endy” is defined in the TOP\_IO & BOTTOM\_IO section.

Template xml file [https://github.com/SymbiFlow/symbiflow-arch-defs/blob/master/quicklogic/qlf\\_k4n8/devices/umc22/interface-mapping\\_24x24.xml](https://github.com/SymbiFlow/symbiflow-arch-defs/blob/master/quicklogic/qlf_k4n8/devices/umc22/interface-mapping_24x24.xml) for a 24x24 device corresponding vpr\_arch xml looks like the following:

```
<?xml version="1.0" encoding="utf-8"?>

<DEVICE name= "qlf_k4n8_umc22" family="qlf_k4n8" width="26" height="26" z="16">
  <IO>
```

(continues on next page)

(continued from previous page)

```

        <TOP_IO y="25">
            <CELL port_name="f2a_i" mapped_name="gfpga_pad_IO_F2A[0:383]" startx="1"
↪endx="24"/>
            <CELL port_name="a2f_o" mapped_name="gfpga_pad_IO_A2F[0:383]" startx="1"
↪endx="24"/>
        </TOP_IO>
        <RIGHT_IO x="25">
            <CELL port_name="f2a_i" mapped_name="gfpga_pad_IO_F2A[384:767]" starty="24"
↪endy="1"/>
            <CELL port_name="a2f_o" mapped_name="gfpga_pad_IO_A2F[384:767]" starty="24"
↪endy="1"/>
        </RIGHT_IO>
        <BOTTOM_IO y="0">
            <CELL port_name="f2a_i" mapped_name="gfpga_pad_IO_F2A[768:1151]" startx="24"
↪endx="1"/>
            <CELL port_name="a2f_o" mapped_name="gfpga_pad_IO_A2F[768:1151]" startx="24"
↪endx="1"/>
        </BOTTOM_IO>
        <LEFT_IO x="0">
            <CELL port_name="f2a_i" mapped_name="gfpga_pad_IO_F2A[1152:1535]" starty="1"
↪endy="24"/>
            <CELL port_name="a2f_o" mapped_name="gfpga_pad_IO_A2F[1152:1535]" starty="1"
↪endy="24"/>
        </LEFT_IO>
    </IO>
</DEVICE>

```

**Note:** When more than one port is specified for mapping at a particular location, then the user has a choice to choose any one of the ports, in csv file specification. It is an error if pin mapping is specified for more than one port at a particular location (same x, y and z coordinates).

## CSV File Specification

Symbiflow is going to dump a template csv file which eFPGA users can update to specify their pin-mapping specifications.

CSV file contains the following column (in the specified order) and mandatory options must be specified.

1. **Orientation:** Valid values are – TOP, BOTTOM, RIGHT or LEFT. Pre-filled in the template file dumped by Symbiflow. **Mandatory option.**
2. **Row:** Row number of the IO cell containing this pin. Pre-filled in the template file dumped by Symbiflow.
3. **Column:** Column number of the IO cell containing this pin. Pre-filled in the template file dumped by Symbiflow.
4. **Pin number in the cell:** Pin number in the cell. Pre-filled in the template file dumped by Symbiflow.
5. **Port name:** Port name (can be scalar, bus or part-select) defined at interface cell-level. Pre-filled in the template file dumped by Symbiflow. **Mandatory option.**
6. **Mapped pin name:** User-defined mapped pin-name. If it is not defined, then the default value for the output port is “NA” and for the input port is “GND”. Needs to be entered by the user for whichever interface port they need to map with. User can edit the interface port name as per their pin specification i.e. if bus interface port



name is specified and user wants to specify pin-mapping only for a single scalar interface port from the given bus port, then user can edit the port name in that particular row in the csv file. **For GPIO pins, user must specify an index like 0, 1 etc, instead of specifying any text name**

7. **GPIO pin type:** Specify “GPIO\_IN”, “GPIO\_OUT” or “GPIO\_EN” to define if the particular port is mapped to a general purpose IO (GPIO) of either of these types - IN (input), OUT (output) or EN (enable). In IO fix placement constraints (defined via input pcf file in symbiflow) can only be defined on the GPIO pins. Default value is “No”.
8. **Associated Clock:** Specify device clock associated to this port. In case of qlf\_k4n8 device, device clocks are - CLK0, CLK1, CLK2 and CLK3. You can specify multiple clocks with a space in between.
9. **Clock Edge:** Type of clock edge at which the data is available. The value can be rising or falling.
10. **Clock Orientation:** CLK0 -> BOTTOM, CLK1 -> RIGHT, CLK2 -> TOP, CLK3 -> LEFT.

### Points to Note

1. In case multiple ports are defined in an xml file for a particular location, template csv file contains one port out of those specified ports at a particular location. Users can choose mapping for that port specified in the csv file OR may choose another port from the xml file for specifying a pin-mapping at that location.
2. It is an error if more than one port is specified in a csv file at a specific location (same x, y and z coordinates).
3. If a user specifies bus-port specification in the csv file for pin-mapping, then it is not required to specify row, column or pin\_num\_in\_cell for that bus-port.
4. If a user specifies scalar-port specification in the csv file for pin-mapping, then also it is not mandatory to specify row, column or pin\_num\_in\_cell. They are mentioned in the template csv files for users to visualize the port location and decide on pin-mapping.
5. Users must specify package name as the csv file name i.e. in file <PACKAGE>.csv, <PACKAGE> is considered as the package name for the pin-mapping specification provided in this csv file.

Template csv file that Symbiflow is going to dump out for a 32x32 looks like the following:

**Note:** At a specific location either A2F or F2A signal can be mapped but not both. For example, both gfpaga\_pad\_IO\_F2A[0] (output port) & gfpaga\_pad\_IO\_A2F[0] (input port) cannot have pin-mapping defined. An error is reported if multiple port mappings specified at a specific location.

Below is the generated template csv file for a 4x4 device. It contains a gfpaga\_pad\_IO\_A2F port but users can alternatively use gfpaga\_pad\_IO\_F2A at any location for pin-mapping.

**orientation,row,col,pin\_num\_in\_cell,port\_name,mapped\_pin,GPIO\_type,Associated Clock,Clock Edge**

```
TOP,5,1,0,gfpaga_pad_IO_A2F[0],,,,
TOP,5,1,1,gfpaga_pad_IO_A2F[1],,,,
TOP,5,2,0,gfpaga_pad_IO_A2F[2],,,,
TOP,5,2,1,gfpaga_pad_IO_A2F[3],,,,
TOP,5,3,0,gfpaga_pad_IO_A2F[4],,,,
TOP,5,3,1,gfpaga_pad_IO_A2F[5],,,,
TOP,5,4,0,gfpaga_pad_IO_A2F[6],,,,
TOP,5,4,1,gfpaga_pad_IO_A2F[7],,,,
BOTTOM,0,4,0,gfpaga_pad_IO_A2F[128],,,,
BOTTOM,0,4,1,gfpaga_pad_IO_A2F[129],,,,
```

(continues on next page)

(continued from previous page)

```

BOTTOM,0,3,0,gfpga_pad_IO_A2F[130],,,,
BOTTOM,0,3,1,gfpga_pad_IO_A2F[131],,,,
BOTTOM,0,2,0,gfpga_pad_IO_A2F[132],,,,
BOTTOM,0,2,1,gfpga_pad_IO_A2F[133],,,,
BOTTOM,0,1,0,gfpga_pad_IO_A2F[134],,,,
BOTTOM,0,1,1,gfpga_pad_IO_A2F[135],,,,
LEFT,1,0,0,gfpga_pad_IO_A2F[192],,,,
LEFT,1,0,1,gfpga_pad_IO_A2F[193],,,,
LEFT,2,0,0,gfpga_pad_IO_A2F[194],,,,
LEFT,2,0,1,gfpga_pad_IO_A2F[195],,,,
RIGHT,4,5,0,gfpga_pad_IO_A2F[64],,,,
RIGHT,4,5,1,gfpga_pad_IO_A2F[65],,,,
RIGHT,3,5,0,gfpga_pad_IO_A2F[66],,,,
RIGHT,3,5,1,gfpga_pad_IO_A2F[67],,,,
RIGHT,2,5,0,gfpga_pad_IO_A2F[68],,,,
RIGHT,2,5,1,gfpga_pad_IO_A2F[69],,,,
RIGHT,1,5,0,gfpga_pad_IO_A2F[70],,,,
RIGHT,1,5,1,gfpga_pad_IO_A2F[71],,,,

```

Users can take the above mentioned csv file and update it in the following manner to define their pin-mapping. Users can specify only those rows where a pin-mapping is specified. It is optional to specify other rows where no pin-mapping is specified. CSV file named as: 'PACK\_4x4.csv' is defined as follows:

**orientation,row,col,pin\_num\_in\_cell,port\_name,mapped\_pin,GPIO\_type,Associated Clock,Clock Edge**

```

TOP,,,gfpga_pad_IO_F2A[1:4],user_out_T[0:3],,CLK0,
TOP,5,3,1,gfpga_pad_IO_A2F[5],0,GPIO_IN,,
TOP,5,4,0,gfpga_pad_IO_F2A[6],0,GPIO_OUT,,
TOP,5,4,1,gfpga_pad_IO_F2A[7],0,GPIO_EN,,
BOTTOM,,,gfpga_pad_IO_F2A[16:18],user_out_B[2:0],,CLK2 CLK3,
BOTTOM,0,2,0,gfpga_pad_IO_A2F[132],1,GPIO_IN,,
BOTTOM,0,2,1,gfpga_pad_IO_F2A[133],1,GPIO_OUT,,
BOTTOM,0,1,0,gfpga_pad_IO_F2A[134],1,GPIO_EN,,
RIGHT,4,5,0,gfpga_pad_IO_A2F[64],user_in_R[0],,,
RIGHT,3,5,0,gfpga_pad_IO_A2F[66],user_in_R[1],,CLK1,

```

**Note:** > The pinmap and csv files are only supported for the eFPGA flow for **QLF\_K4N8** > In the above example, the first row represents the pin-mapping with bus-ports. > In this row, gfpga\_pad\_IO\_F2A[1:4] is mapped to user-defined pins: user\_out\_T[0:3] such that gfpga\_pad\_IO\_F2A[1] is mapped to user\_out\_T[0], gfpga\_pad\_IO\_F2A[2] is mapped to user\_out\_T[1] and so on.

## SDC File Specification

In case of eFPGA flow, the user needs to provide SDC timing constraints on the mapped pin name/net name.

SDC File can be specified as input with timing constraints applied on the mapped user-defined pins.

Following are the SDC commands used for specifying the I/O constraints.

You can read about these commands in more detail at: [https://docs.verilogtorouting.org/en/latest/vpr/sdc\\_commands/](https://docs.verilogtorouting.org/en/latest/vpr/sdc_commands/)

### create\_clock

This constraint creates a design clock and defines its characteristics. Clock characteristics include clock name, clock period, waveform, and clock source.

#### Syntax:

```
create_clock -name clockName -period period_float_values [-waveform edge_list]
```

#### Example:

```
create_clock -name CLK -period 2.0 -waveform {0 5}
```

This example generates a clock named CLK, whose clock period is 2.0ns and the clock source is available at the clk port. The clock edges are 0.0 and 1.0, respectively.

### set\_input\_delay/set\_output\_delay

Use *set\_input\_delay* if you want timing paths from input I/Os analyzed, and *set\_output\_delay* if you want timing paths to output I/Os analyzed.

These commands constrain each I/O pad specified after *get\_ports* to be timing-equivalent to a register clocked on the clock specified after *-clock*. This can be either a clock signal in your design or a virtual clock that does not exist in the design but which is used only to specify the timing of I/Os.

The specified delays are added to I/O timing paths and can be used to model board level delays.

This constraint sets the external minimum or maximum arrival time for the design or device input pin with respect to the specified reference clock. This constraint can be used to perform timing analysis from an external source to the next sequential element that is in eFPGA. Since the element is in eFPGA, the user can constrain the design at the eFPGA input.

#### Syntax:

```
set_input_delay delay_float_value -clock ref_clock [-max] [-min] [-clock_fall]   
↪input_port/pin_list
```

#### Example:

```
set_input_delay 2.0 -max [get_ports {IN}]
```

This example sets the input delay of 2.0 ns at the default input port and sets the maximum delay.

#### Syntax:

```
set_output_delay delay_float_value -clock ref_clock [-max] [-min] [-clock_
↪fall] output_port_list
```

**Example:**

```
set_output_delay 1.0 -max [get_ports {count[0]}]
```

This example sets the output delay to 1.0ns at the count[0] port and sets the maximum delay.

**Sample SDC File**

Sample SDC file looks like the following:

```
create_clock -name SYS_CLK_0 -period 10 -waveform {0 5}
create_clock -name SYS_CLK_1 -period 10 -waveform {0 5}
create_clock -name SYS_CLK_2 -period 10 -waveform {0 5}
create_clock -name SYS_CLK_3 -period 10 -waveform {0 5}
create_clock -name SYS_CLK_4 -period 10 -waveform {0 5}
set_output_delay 10 -max -clock SYS_CLK_2 [get_ports F1]
set_output_delay -0 -min -clock SYS_CLK_2 [get_ports F1]
set_input_delay 10 -max -clock SYS_CLK_2 [get_ports A1]
set_input_delay 0 -min -clock SYS_CLK_2 [get_ports A1]
set_output_delay 10 -max -clock SYS_CLK_0 [get_ports F2]
set_output_delay -0 -min -clock SYS_CLK_0 [get_ports F2]
set_input_delay 10 -max -clock SYS_CLK_1 [get_ports A2]
set_input_delay 0 -min -clock SYS_CLK_1 [get_ports A2]
```

**2.3.4 Running Pack, Place and Route Tools**

The eblif file generated during the synthesis is used for pack, place and route along with device information, pcf and the sdc file. User needs to provide the csv, sdc and pcf file. These are optional input files. CSV and pcf file is not present in the test folder.

Syntax:

```
ql_symbiflow -compile -src <source complete path> -d <device> -t <top_
↪module name> -v <verilog files> -p <pcf file> -P <Package CSV file> -
↪s <SDC file> -pnr_corner <fast/slow>
```

The output files dumped will be: <TOP>.net : Once packing is complete. <TOP>.place : Placer file from VPR <TOP>.route : Router file from VPR

One can refer to the pack.log, placer.log, router.log for more information related to each tool.

```
cd <INSTALL_PATH>/quicklogic-arch-defs/share/symbiflow/tests/counter_
↪16bit
ql_symbiflow -compile -src $PWD -d qlf_k4n8 -t top -v counter_16bit.v -
↪s counter_16bit.sdc
```

The above command will also run synthesis if it was not run before.

If user wants to do IO fix placement, then pcf and csv files need to be specified in the following manner:

```
cd <INSTALL_PATH>/quicklogic-arch-defs/share/symbiflow/tests/counter_
↳16bit
ql_symbiflow -compile -src $PWD -d qlf_k4n8 -t top -v counter_16bit.v -
↳s counter_16bit.sdc -p counter_16bit.pcf -P pinmap_qlf_k4n8_umc22.
↳csv -pnr_corner fast
```

The Timing analysis refer the files *report\_timing.hold.rpt*, *report\_timing.setup.rpt* and *top.log* inside the build folder

For the counter design below is the timing report from the top.log file:

```
Hold Worst Negative Slack (hWNS):    0 ns
Hold Total Negative Slack (hTNS):    0 ns
Setup Worst Negative Slack (sWNS):   -2.91 ns
Setup Total Negative Slack (sTNS):   -162.424 ns
Final critical path: 2.91 ns, Fmax:  343.643 MHz
```

---

**Note:** Options : -P, -s, -p, -pnr\_corner are only available for **QLF\_K4N8**

---

### 2.3.5 Performing the Post-Layout Timing Simulation

Post layout Timing simulation uses the SDF(Standard Delay Format) file.

The testbench for the counter design is present at: *<Install\_Path>/quicklogic-arch-defs/share/symbiflow/tests/counter\_16bit/*

The post-layout design netlist is present at: *<Install\_Path>/quicklogic-arch-defs/share/symbiflow/counter\_16bit/build/top\_post\_synthesis.v*

The SDF file is present at: *<Install\_Path>/quicklogic-arch-defs/share/symbiflow/counter\_16bit/build/top\_post\_synthesis.sdf*

The primitive file library file is present at: *<Install\_Path>/quicklogic-arch-defs/share/symbiflow/techmaps/quicklogic/techmaps/<Family>/cells\_sim.v*

**To perform a post-layout simulation:** - Perform a post-layout simulation of the Verilog code use iverilog. - View the simulation results in the Waveform/ Data Analyzer and verify.

---

**Note:** > cells\_sim.v : This file has the definition for technology mapped macros > Supported only for **QLF\_K4N8**

---

### 2.3.6 Generate the Programming Files

The technology mapped netlist and packing/placement/routing results produced by VPR contain the information needed to generate a device programming bitstreams. These bitstreams can be used to configure the qlf\_k4n8 eFPGA device for validation:

1. Bitstream File
2. FourByte File

## Bitstream File

This programming file is the binary bitstream file with extension <top module>.bin. It will be generated by default after the PnR run by Symbiflow.

## Four Byte File

This programming file is the 4Byte/hex file with extension <top module>.bit. It will be generated by default post PnR run by Symbiflow. This file can be used for validating the designs. To run in the test-bench -mode requires changes. Refer README of the test-bench package.

---

**Note:** > The programming files are currently supported only for the device **QLF\_K4N8**

---

## 2.3.7 PCF Sample

The PCF file is for fix placing the IO to a particular IO location on the device. For qlf\_k4n8 device, user needs to create a pin mapping csv (as defined in Pin Mapping section). The user-defined pin names can then be specified in the pcf file for IO placement. Below mentioned CSV file(not part of package) corresponds to a pin-mapping defined for 24x24 device size for qlf\_k4n8.

### 24x24 Device Pin Mapping CSV File

To map the IO's in the pcf file, refer the below csv file. The column mapped\_pin names are used to map in the pcf file. For example if a design has two input ports A\_in, B\_in and one output port C\_out, the pcf file will look like this.

```
set_io A_in A2F_GPIO2_29
set_io B_in A2F_GPIO2_28
set_io C_out F2A_GPIO2_31
```

**If a user does not provide a pcf file, then one can refer the <design>.place file.**

Where mapping from <design>.place file to csv file is as shown:

```
x -> col
y -> row
subblk -> pin_number
```

```
orientation,row,col,pin_num_in_cell,port_name,mapped_pin,GPIO_type
Top,0,23,15,gfpga_pad_IO_F2A[383],F2A_GPIO2_31,
Top,0,23,14,gfpga_pad_IO_F2A[382],F2A_GPIO2_30,
Top,0,23,13,gfpga_pad_IO_F2A[381],F2A_GPIO2_29,
Top,0,23,12,gfpga_pad_IO_F2A[380],F2A_GPIO2_28,
Top,0,23,11,gfpga_pad_IO_A2F[379],A2F_GPIO2_31,
Top,0,23,10,gfpga_pad_IO_A2F[378],A2F_GPIO2_30,
Top,0,23,9,gfpga_pad_IO_A2F[377],A2F_GPIO2_29,
Top,0,23,8,gfpga_pad_IO_A2F[376],A2F_GPIO2_28,
Top,0,22,15,gfpga_pad_IO_F2A[367],F2A_GPIO2_27,
Top,0,22,14,gfpga_pad_IO_F2A[366],F2A_GPIO2_26,
Top,0,22,13,gfpga_pad_IO_F2A[365],F2A_GPIO2_25,
Top,0,22,12,gfpga_pad_IO_F2A[364],F2A_GPIO2_24,
```

(continues on next page)

(continued from previous page)

```

Top,0,22,11,gfpga_pad_IO_A2F[363],A2F_GPIO2_27,
Top,0,22,10,gfpga_pad_IO_A2F[362],A2F_GPIO2_26,
Top,0,22,9,gfpga_pad_IO_A2F[361],A2F_GPIO2_25,
Top,0,22,8,gfpga_pad_IO_A2F[360],A2F_GPIO2_24,
Top,0,21,15,gfpga_pad_IO_F2A[351],F2A_GPIO2_23,
Top,0,21,14,gfpga_pad_IO_F2A[350],F2A_GPIO2_22,
Top,0,21,13,gfpga_pad_IO_F2A[349],F2A_GPIO2_21,
Top,0,21,12,gfpga_pad_IO_F2A[348],F2A_GPIO2_20,
Top,0,21,11,gfpga_pad_IO_A2F[347],A2F_GPIO2_23,
Top,0,21,10,gfpga_pad_IO_A2F[346],A2F_GPIO2_22,
Top,0,21,9,gfpga_pad_IO_A2F[345],A2F_GPIO2_21,
Top,0,21,8,gfpga_pad_IO_A2F[344],A2F_GPIO2_20,
Top,0,21,0,gfpga_pad_IO_A2F[336],A2F_RTC_8KHZ,
Top,0,20,15,gfpga_pad_IO_F2A[335],F2A_GPIO2_19,
Top,0,20,14,gfpga_pad_IO_F2A[334],F2A_GPIO2_18,
Top,0,20,13,gfpga_pad_IO_F2A[333],F2A_GPIO2_17,
Top,0,20,12,gfpga_pad_IO_F2A[332],F2A_GPIO2_16,
Top,0,20,11,gfpga_pad_IO_A2F[331],A2F_GPIO2_19,
Top,0,20,10,gfpga_pad_IO_A2F[330],A2F_GPIO2_18,
Top,0,20,9,gfpga_pad_IO_A2F[329],A2F_GPIO2_17,
Top,0,20,8,gfpga_pad_IO_A2F[328],A2F_GPIO2_16,
Top,0,20,1,gfpga_pad_IO_A2F[321],A2F_USB_CLK_EVENT,
Top,0,20,0,gfpga_pad_IO_F2A[320],F2A_EEC_IN,
Top,0,19,15,gfpga_pad_IO_F2A[319],F2A_GPIO2_15,
Top,0,19,14,gfpga_pad_IO_F2A[318],F2A_GPIO2_14,
Top,0,19,13,gfpga_pad_IO_F2A[317],F2A_GPIO2_13,
Top,0,19,12,gfpga_pad_IO_F2A[316],F2A_GPIO2_12,
Top,0,19,11,gfpga_pad_IO_A2F[315],A2F_GPIO2_15,
Top,0,19,10,gfpga_pad_IO_A2F[314],A2F_GPIO2_14,
Top,0,19,9,gfpga_pad_IO_A2F[313],A2F_GPIO2_13,
Top,0,19,8,gfpga_pad_IO_A2F[312],A2F_GPIO2_12,
Top,0,19,5,gfpga_pad_IO_A2F[309],RFU,
Top,0,19,4,gfpga_pad_IO_A2F[308],RFU,
Top,0,19,3,gfpga_pad_IO_F2A[307],RFU,
Top,0,19,2,gfpga_pad_IO_F2A[306],RFU,
Top,0,19,1,gfpga_pad_IO_A2F[305],A2F_IRQ9_ACK,
Top,0,19,0,gfpga_pad_IO_F2A[304],F2A_IRQ9,
Top,0,18,15,gfpga_pad_IO_F2A[303],F2A_GPIO2_11,
Top,0,18,14,gfpga_pad_IO_F2A[302],F2A_GPIO2_10,
Top,0,18,13,gfpga_pad_IO_F2A[301],F2A_GPIO2_9,
Top,0,18,12,gfpga_pad_IO_F2A[300],F2A_GPIO2_8,
Top,0,18,11,gfpga_pad_IO_A2F[299],A2F_GPIO2_11,
Top,0,18,10,gfpga_pad_IO_A2F[298],A2F_GPIO2_10,
Top,0,18,9,gfpga_pad_IO_A2F[297],A2F_GPIO2_9,
Top,0,18,8,gfpga_pad_IO_A2F[296],A2F_GPIO2_8,
Top,0,18,5,gfpga_pad_IO_F2A[293],F2A_TSC9_EVENT,
Top,0,18,4,gfpga_pad_IO_F2A[292],F2A_TSC8_EVENT,
Top,0,18,3,gfpga_pad_IO_F2A[291],RFU,
Top,0,18,2,gfpga_pad_IO_F2A[290],RFU,
Top,0,18,1,gfpga_pad_IO_A2F[289],A2F_IRQ8_ACK,
Top,0,18,0,gfpga_pad_IO_F2A[288],F2A_IRQ8,
Top,0,17,15,gfpga_pad_IO_F2A[287],F2A_GPIO2_7,

```

(continues on next page)

(continued from previous page)

```

Top,0,17,14,gfpga_pad_IO_F2A[286],F2A_GPIO2_6,
Top,0,17,13,gfpga_pad_IO_F2A[285],F2A_GPIO2_5,
Top,0,17,12,gfpga_pad_IO_F2A[284],F2A_GPIO2_4,
Top,0,17,11,gfpga_pad_IO_A2F[283],A2F_GPIO2_7,
Top,0,17,10,gfpga_pad_IO_A2F[282],A2F_GPIO2_6,
Top,0,17,9,gfpga_pad_IO_A2F[281],A2F_GPIO2_5,
Top,0,17,8,gfpga_pad_IO_A2F[280],A2F_GPIO2_4,
Top,0,17,4,gfpga_pad_IO_F2A[276],F2A_TSC6_EVENT,
Top,0,17,3,gfpga_pad_IO_A2F[275],A2F_GPIO2_WR_ACK,
Top,0,17,2,gfpga_pad_IO_A2F[274],A2F_GPIO2_RD_ACK,
Top,0,17,1,gfpga_pad_IO_A2F[273],A2F_IRQ6_ACK,
Top,0,17,0,gfpga_pad_IO_F2A[272],F2A_IRQ6,
Top,0,16,15,gfpga_pad_IO_F2A[271],F2A_GPIO2_3,
Top,0,16,14,gfpga_pad_IO_F2A[270],F2A_GPIO2_2,
Top,0,16,13,gfpga_pad_IO_F2A[269],F2A_GPIO2_1,
Top,0,16,12,gfpga_pad_IO_F2A[268],F2A_GPIO2_0,
Top,0,16,11,gfpga_pad_IO_A2F[267],A2F_GPIO2_3,
Top,0,16,10,gfpga_pad_IO_A2F[266],A2F_GPIO2_2,
Top,0,16,9,gfpga_pad_IO_A2F[265],A2F_GPIO2_1,
Top,0,16,8,gfpga_pad_IO_A2F[264],A2F_GPIO2_0,
Top,0,16,5,gfpga_pad_IO_F2A[261],F2A_TSC5_EVENT,
Top,0,16,4,gfpga_pad_IO_F2A[260],F2A_TSC4_EVENT,
Top,0,16,3,gfpga_pad_IO_A2F[259],A2F_GPIO2_WR_ACK,
Top,0,16,2,gfpga_pad_IO_A2F[258],A2F_GPIO2_RD_ACK,
Top,0,16,1,gfpga_pad_IO_A2F[257],A2F_IRQ5_ACK,
Top,0,16,0,gfpga_pad_IO_F2A[256],F2A_IRQ5,
Top,0,15,15,gfpga_pad_IO_F2A[255],F2A_GPIO1_31,
Top,0,15,14,gfpga_pad_IO_F2A[254],F2A_GPIO1_30,
Top,0,15,13,gfpga_pad_IO_F2A[253],F2A_GPIO1_29,
Top,0,15,12,gfpga_pad_IO_F2A[252],F2A_GPIO1_28,
Top,0,15,11,gfpga_pad_IO_A2F[251],A2F_GPIO1_31,
Top,0,15,10,gfpga_pad_IO_A2F[250],A2F_GPIO1_30,
Top,0,15,9,gfpga_pad_IO_A2F[249],A2F_GPIO1_29,
Top,0,15,8,gfpga_pad_IO_A2F[248],A2F_GPIO1_28,
Top,0,15,5,gfpga_pad_IO_A2F[245],RFU,
Top,0,15,4,gfpga_pad_IO_A2F[244],RFU,
Top,0,15,3,gfpga_pad_IO_F2A[243],RFU,
Top,0,15,2,gfpga_pad_IO_F2A[242],RFU,
Top,0,15,1,gfpga_pad_IO_F2A[241],F2A_RAM_WE1_HS,
Top,0,15,0,gfpga_pad_IO_F2A[240],F2A_RAM_WE0_HS,
Top,0,14,15,gfpga_pad_IO_F2A[239],F2A_GPIO1_27,
Top,0,14,14,gfpga_pad_IO_F2A[238],F2A_GPIO1_26,
Top,0,14,13,gfpga_pad_IO_F2A[237],F2A_GPIO1_25,
Top,0,14,12,gfpga_pad_IO_F2A[236],F2A_GPIO1_24,
Top,0,14,11,gfpga_pad_IO_A2F[235],A2F_GPIO1_27,
Top,0,14,10,gfpga_pad_IO_A2F[234],A2F_GPIO1_26,
Top,0,14,9,gfpga_pad_IO_A2F[233],A2F_GPIO1_25,
Top,0,14,8,gfpga_pad_IO_A2F[232],A2F_GPIO1_24,
Top,0,14,7,gfpga_pad_IO_F2A[231],F2A_RAM_DATA15_HS,
Top,0,14,6,gfpga_pad_IO_F2A[230],F2A_RAM_DATA14_HS,
Top,0,14,5,gfpga_pad_IO_F2A[229],F2A_RAM_DATA13_HS,
Top,0,14,4,gfpga_pad_IO_F2A[228],F2A_RAM_DATA12_HS,

```

(continues on next page)



(continued from previous page)

```

Top,0,14,3,gfpga_pad_IO_F2A[227],F2A_RAM_DATA11_HS,
Top,0,14,2,gfpga_pad_IO_F2A[226],F2A_RAM_DATA10_HS,
Top,0,14,1,gfpga_pad_IO_F2A[225],F2A_RAM_DATA9_HS,
Top,0,14,0,gfpga_pad_IO_F2A[224],F2A_RAM_DATA8_HS,
Top,0,13,15,gfpga_pad_IO_F2A[223],F2A_GPIO1_23,
Top,0,13,14,gfpga_pad_IO_F2A[222],F2A_GPIO1_22,
Top,0,13,13,gfpga_pad_IO_F2A[221],F2A_GPIO1_21,
Top,0,13,12,gfpga_pad_IO_F2A[220],F2A_GPIO1_20,
Top,0,13,11,gfpga_pad_IO_A2F[219],A2F_GPIO1_23,
Top,0,13,10,gfpga_pad_IO_A2F[218],A2F_GPIO1_22,
Top,0,13,9,gfpga_pad_IO_A2F[217],A2F_GPIO1_21,
Top,0,13,8,gfpga_pad_IO_A2F[216],A2F_GPIO1_20,
Top,0,13,7,gfpga_pad_IO_F2A[215],F2A_RAM_DATA7_HS,
Top,0,13,6,gfpga_pad_IO_F2A[214],F2A_RAM_DATA6_HS,
Top,0,13,5,gfpga_pad_IO_F2A[213],F2A_RAM_DATA5_HS,
Top,0,13,4,gfpga_pad_IO_F2A[212],F2A_RAM_DATA4_HS,
Top,0,13,3,gfpga_pad_IO_F2A[211],F2A_RAM_DATA3_HS,
Top,0,13,2,gfpga_pad_IO_F2A[210],F2A_RAM_DATA2_HS,
Top,0,13,1,gfpga_pad_IO_F2A[209],F2A_RAM_DATA1_HS,
Top,0,13,0,gfpga_pad_IO_F2A[208],F2A_RAM_DATA0_HS,
Top,0,12,15,gfpga_pad_IO_F2A[207],F2A_GPIO1_19,
Top,0,12,14,gfpga_pad_IO_F2A[206],F2A_GPIO1_18,
Top,0,12,13,gfpga_pad_IO_F2A[205],F2A_GPIO1_17,
Top,0,12,12,gfpga_pad_IO_F2A[204],F2A_GPIO1_16,
Top,0,12,11,gfpga_pad_IO_A2F[203],A2F_GPIO1_19,
Top,0,12,10,gfpga_pad_IO_A2F[202],A2F_GPIO1_18,
Top,0,12,9,gfpga_pad_IO_A2F[201],A2F_GPIO1_17,
Top,0,12,8,gfpga_pad_IO_A2F[200],A2F_GPIO1_16,
Top,0,12,6,gfpga_pad_IO_F2A[198],F2A_RAM_ADDR14_HS,
Top,0,12,5,gfpga_pad_IO_F2A[197],F2A_RAM_ADDR13_HS,
Top,0,12,4,gfpga_pad_IO_F2A[196],F2A_RAM_ADDR12_HS,
Top,0,12,3,gfpga_pad_IO_F2A[195],F2A_RAM_ADDR11_HS,
Top,0,12,2,gfpga_pad_IO_F2A[194],F2A_RAM_ADDR10_HS,
Top,0,12,1,gfpga_pad_IO_F2A[193],F2A_RAM_ADDR9_HS,
Top,0,12,0,gfpga_pad_IO_F2A[192],F2A_RAM_ADDR8_HS,
Top,0,11,15,gfpga_pad_IO_F2A[191],F2A_GPIO1_15,
Top,0,11,14,gfpga_pad_IO_F2A[190],F2A_GPIO1_14,
Top,0,11,13,gfpga_pad_IO_F2A[189],F2A_GPIO1_13,
Top,0,11,12,gfpga_pad_IO_F2A[188],F2A_GPIO1_12,
Top,0,11,11,gfpga_pad_IO_A2F[187],A2F_GPIO1_15,
Top,0,11,10,gfpga_pad_IO_A2F[186],A2F_GPIO1_14,
Top,0,11,9,gfpga_pad_IO_A2F[185],A2F_GPIO1_13,
Top,0,11,8,gfpga_pad_IO_A2F[184],A2F_GPIO1_12,
Top,0,11,7,gfpga_pad_IO_F2A[183],F2A_RAM_ADDR7_HS,
Top,0,11,6,gfpga_pad_IO_F2A[182],F2A_RAM_ADDR6_HS,
Top,0,11,5,gfpga_pad_IO_F2A[181],F2A_RAM_ADDR5_HS,
Top,0,11,4,gfpga_pad_IO_F2A[180],F2A_RAM_ADDR4_HS,
Top,0,11,3,gfpga_pad_IO_F2A[179],F2A_RAM_ADDR3_HS,
Top,0,11,2,gfpga_pad_IO_F2A[178],F2A_RAM_ADDR2_HS,
Top,0,11,1,gfpga_pad_IO_F2A[177],F2A_RAM_ADDR1_HS,
Top,0,11,0,gfpga_pad_IO_F2A[176],F2A_RAM_CE_HS,
Top,0,10,15,gfpga_pad_IO_F2A[175],F2A_GPIO1_11,

```

(continues on next page)

(continued from previous page)

```

Top,0,10,14,gfpga_pad_IO_F2A[174],F2A_GPIO1_10,
Top,0,10,13,gfpga_pad_IO_F2A[173],F2A_GPIO1_9,
Top,0,10,12,gfpga_pad_IO_F2A[172],F2A_GPIO1_8,
Top,0,10,11,gfpga_pad_IO_A2F[171],A2F_GPIO1_11,
Top,0,10,10,gfpga_pad_IO_A2F[170],A2F_GPIO1_10,
Top,0,10,9,gfpga_pad_IO_A2F[169],A2F_GPIO1_9,
Top,0,10,8,gfpga_pad_IO_A2F[168],A2F_GPIO1_8,
Top,0,10,7,gfpga_pad_IO_A2F[167],A2F_RAM_DATA7_HS,
Top,0,10,6,gfpga_pad_IO_A2F[166],A2F_RAM_DATA6_HS,
Top,0,10,5,gfpga_pad_IO_A2F[165],A2F_RAM_DATA5_HS,
Top,0,10,4,gfpga_pad_IO_A2F[164],A2F_RAM_DATA4_HS,
Top,0,10,3,gfpga_pad_IO_A2F[163],A2F_RAM_DATA3_HS,
Top,0,10,2,gfpga_pad_IO_A2F[162],A2F_RAM_DATA2_HS,
Top,0,10,1,gfpga_pad_IO_A2F[161],A2F_RAM_DATA1_HS,
Top,0,10,0,gfpga_pad_IO_A2F[160],A2F_RAM_DATA0_HS,
Top,0,9,15,gfpga_pad_IO_F2A[159],F2A_GPIO1_7,
Top,0,9,14,gfpga_pad_IO_F2A[158],F2A_GPIO1_6,
Top,0,9,13,gfpga_pad_IO_F2A[157],F2A_GPIO1_5,
Top,0,9,12,gfpga_pad_IO_F2A[156],F2A_GPIO1_4,
Top,0,9,11,gfpga_pad_IO_A2F[155],A2F_GPIO1_7,
Top,0,9,10,gfpga_pad_IO_A2F[154],A2F_GPIO1_6,
Top,0,9,9,gfpga_pad_IO_A2F[153],A2F_GPIO1_5,
Top,0,9,8,gfpga_pad_IO_A2F[152],A2F_GPIO1_4,
Top,0,9,7,gfpga_pad_IO_A2F[151],A2F_RAM_DATA15_HS,
Top,0,9,6,gfpga_pad_IO_A2F[150],A2F_RAM_DATA14_HS,
Top,0,9,5,gfpga_pad_IO_A2F[149],A2F_RAM_DATA13_HS,
Top,0,9,4,gfpga_pad_IO_A2F[148],A2F_RAM_DATA12_HS,
Top,0,9,3,gfpga_pad_IO_A2F[147],A2F_RAM_DATA11_HS,
Top,0,9,2,gfpga_pad_IO_A2F[146],A2F_RAM_DATA10_HS,
Top,0,9,1,gfpga_pad_IO_A2F[145],A2F_RAM_DATA9_HS,
Top,0,9,0,gfpga_pad_IO_A2F[144],A2F_RAM_DATA8_HS,
Top,0,8,15,gfpga_pad_IO_F2A[143],F2A_GPIO1_3,
Top,0,8,14,gfpga_pad_IO_F2A[142],F2A_GPIO1_2,
Top,0,8,13,gfpga_pad_IO_F2A[141],F2A_GPIO1_1,
Top,0,8,12,gfpga_pad_IO_F2A[140],F2A_GPIO1_0,
Top,0,8,11,gfpga_pad_IO_A2F[139],A2F_GPIO1_3,
Top,0,8,10,gfpga_pad_IO_A2F[138],A2F_GPIO1_2,
Top,0,8,9,gfpga_pad_IO_A2F[137],A2F_GPIO1_1,
Top,0,8,8,gfpga_pad_IO_A2F[136],A2F_GPIO1_0,
Top,0,8,5,gfpga_pad_IO_F2A[133],F2A_TSC3_EVENT,
Top,0,8,4,gfpga_pad_IO_F2A[132],F2A_TSC2_EVENT,
Top,0,8,3,gfpga_pad_IO_A2F[131],A2F_GPIO0_WR_ACK,
Top,0,8,2,gfpga_pad_IO_A2F[130],A2F_GPIO0_RD_ACK,
Top,0,8,1,gfpga_pad_IO_A2F[129],A2F_IRQ4_ACK,
Top,0,8,0,gfpga_pad_IO_F2A[128],F2A_IRQ4,
Top,0,7,15,gfpga_pad_IO_F2A[127],F2A_GPIO0_31,
Top,0,7,14,gfpga_pad_IO_F2A[126],F2A_GPIO0_30,
Top,0,7,13,gfpga_pad_IO_F2A[125],F2A_GPIO0_29,
Top,0,7,12,gfpga_pad_IO_F2A[124],F2A_GPIO0_28,
Top,0,7,11,gfpga_pad_IO_A2F[123],A2F_GPIO0_31,
Top,0,7,10,gfpga_pad_IO_A2F[122],A2F_GPIO0_30,
Top,0,7,9,gfpga_pad_IO_A2F[121],A2F_GPIO0_29,

```

(continues on next page)

(continued from previous page)

```

Top,0,7,8,gfpga_pad_IO_A2F[120],A2F_GPIO0_28,
Top,0,7,5,gfpga_pad_IO_F2A[117],F2A_TSC1_EVENT,
Top,0,7,4,gfpga_pad_IO_F2A[116],F2A_TSC0_EVENT,
Top,0,7,3,gfpga_pad_IO_A2F[115],A2F_GPIO1_WR_ACK,
Top,0,7,2,gfpga_pad_IO_A2F[114],A2F_GPIO1_RD_ACK,
Top,0,7,1,gfpga_pad_IO_A2F[113],A2F_IRQ3_ACK,
Top,0,7,0,gfpga_pad_IO_F2A[112],F2A_IRQ3,
Top,0,6,15,gfpga_pad_IO_F2A[111],F2A_GPIO0_27,
Top,0,6,14,gfpga_pad_IO_F2A[110],F2A_GPIO0_26,
Top,0,6,13,gfpga_pad_IO_F2A[109],F2A_GPIO0_25,
Top,0,6,12,gfpga_pad_IO_F2A[108],F2A_GPIO0_24,
Top,0,6,11,gfpga_pad_IO_A2F[107],A2F_GPIO0_27,
Top,0,6,10,gfpga_pad_IO_A2F[106],A2F_GPIO0_26,
Top,0,6,9,gfpga_pad_IO_A2F[105],A2F_GPIO0_25,
Top,0,6,8,gfpga_pad_IO_A2F[104],A2F_GPIO0_24,
Top,0,6,5,gfpga_pad_IO_A2F[101],RFU,
Top,0,6,4,gfpga_pad_IO_A2F[100],RFU,
Top,0,6,3,gfpga_pad_IO_F2A[99],RFU,
Top,0,6,2,gfpga_pad_IO_F2A[98],RFU,
Top,0,6,1,gfpga_pad_IO_A2F[97],A2F_IRQ2_ACK,
Top,0,6,0,gfpga_pad_IO_F2A[96],F2A_IRQ2,
Top,0,5,15,gfpga_pad_IO_F2A[95],F2A_GPIO0_23,
Top,0,5,14,gfpga_pad_IO_F2A[94],F2A_GPIO0_22,
Top,0,5,13,gfpga_pad_IO_F2A[93],F2A_GPIO0_21,
Top,0,5,12,gfpga_pad_IO_F2A[92],F2A_GPIO0_20,
Top,0,5,11,gfpga_pad_IO_A2F[91],A2F_GPIO0_23,
Top,0,5,10,gfpga_pad_IO_A2F[90],A2F_GPIO0_22,
Top,0,5,9,gfpga_pad_IO_A2F[89],A2F_GPIO0_21,
Top,0,5,8,gfpga_pad_IO_A2F[88],A2F_GPIO0_20,
Top,0,5,5,gfpga_pad_IO_A2F[85],RFU,
Top,0,5,4,gfpga_pad_IO_A2F[84],RFU,
Top,0,5,3,gfpga_pad_IO_F2A[83],RFU,
Top,0,5,2,gfpga_pad_IO_F2A[82],RFU,
Top,0,5,1,gfpga_pad_IO_A2F[81],A2F_IRQ1_ACK,
Top,0,5,0,gfpga_pad_IO_F2A[80],F2A_IRQ1,
Top,0,4,15,gfpga_pad_IO_F2A[79],F2A_GPIO0_19,
Top,0,4,14,gfpga_pad_IO_F2A[78],F2A_GPIO0_18,
Top,0,4,13,gfpga_pad_IO_F2A[77],F2A_GPIO0_17,
Top,0,4,12,gfpga_pad_IO_F2A[76],F2A_GPIO0_16,
Top,0,4,11,gfpga_pad_IO_A2F[75],A2F_GPIO0_19,
Top,0,4,10,gfpga_pad_IO_A2F[74],A2F_GPIO0_18,
Top,0,4,9,gfpga_pad_IO_A2F[73],A2F_GPIO0_17,
Top,0,4,8,gfpga_pad_IO_A2F[72],A2F_GPIO0_16,
Top,0,4,1,gfpga_pad_IO_A2F[65],A2F_IRQ0_ACK,
Top,0,4,0,gfpga_pad_IO_F2A[64],F2A_IRQ0,
Top,0,3,15,gfpga_pad_IO_F2A[63],F2A_GPIO0_15,
Top,0,3,14,gfpga_pad_IO_F2A[62],F2A_GPIO0_14,
Top,0,3,13,gfpga_pad_IO_F2A[61],F2A_GPIO0_13,
Top,0,3,12,gfpga_pad_IO_F2A[60],F2A_GPIO0_12,
Top,0,3,11,gfpga_pad_IO_A2F[59],A2F_GPIO0_15,
Top,0,3,10,gfpga_pad_IO_A2F[58],A2F_GPIO0_14,
Top,0,3,9,gfpga_pad_IO_A2F[57],A2F_GPIO0_13,

```

(continues on next page)

(continued from previous page)

```

Top,0,3,8,gfpga_pad_IO_A2F[56],A2F_GPIO0_12,
Top,0,3,0,gfpga_pad_IO_A2F[48],A2F_GLOBAL_MUTE,
Top,0,2,15,gfpga_pad_IO_F2A[47],F2A_GPIO0_11,
Top,0,2,14,gfpga_pad_IO_F2A[46],F2A_GPIO0_10,
Top,0,2,13,gfpga_pad_IO_F2A[45],F2A_GPIO0_9,
Top,0,2,12,gfpga_pad_IO_F2A[44],F2A_GPIO0_8,
Top,0,2,11,gfpga_pad_IO_A2F[43],A2F_GPIO0_11,
Top,0,2,10,gfpga_pad_IO_A2F[42],A2F_GPIO0_10,
Top,0,2,9,gfpga_pad_IO_A2F[41],A2F_GPIO0_9,
Top,0,2,8,gfpga_pad_IO_A2F[40],A2F_GPIO0_8,
Top,0,2,0,gfpga_pad_IO_F2A[32],F2A_LB_CLK_T,
Top,0,1,15,gfpga_pad_IO_F2A[31],F2A_GPIO0_7,
Top,0,1,14,gfpga_pad_IO_F2A[30],F2A_GPIO0_6,
Top,0,1,13,gfpga_pad_IO_F2A[29],F2A_GPIO0_5,
Top,0,1,12,gfpga_pad_IO_F2A[28],F2A_GPIO0_4,
Top,0,1,11,gfpga_pad_IO_A2F[27],A2F_GPIO0_7,
Top,0,1,10,gfpga_pad_IO_A2F[26],A2F_GPIO0_6,
Top,0,1,9,gfpga_pad_IO_A2F[25],A2F_GPIO0_5,
Top,0,1,8,gfpga_pad_IO_A2F[24],A2F_GPIO0_4,
Top,0,0,15,gfpga_pad_IO_F2A[15],F2A_GPIO0_3,
Top,0,0,14,gfpga_pad_IO_F2A[14],F2A_GPIO0_2,
Top,0,0,13,gfpga_pad_IO_F2A[13],F2A_GPIO0_1,
Top,0,0,12,gfpga_pad_IO_F2A[12],F2A_GPIO0_0,
Top,0,0,11,gfpga_pad_IO_A2F[11],A2F_GPIO0_3,
Top,0,0,10,gfpga_pad_IO_A2F[10],A2F_GPIO0_2,
Top,0,0,9,gfpga_pad_IO_A2F[9],A2F_GPIO0_1,
Top,0,0,8,gfpga_pad_IO_A2F[8],A2F_GPIO0_0,
Right,20,24,2,gfpga_pad_IO_A2F[706],P2F_GPIO112_IN,
Right,20,24,1,gfpga_pad_IO_F2A[705],F2P_GPIO112_OUT,
Right,20,24,0,gfpga_pad_IO_F2A[704],F2P_GPIO112_OE,
Right,19,24,6,gfpga_pad_IO_A2F[694],P2F_GPIO113_IN,
Right,19,24,5,gfpga_pad_IO_F2A[693],F2P_GPIO113_OUT,
Right,19,24,4,gfpga_pad_IO_F2A[692],F2P_GPIO113_OE,
Right,19,24,2,gfpga_pad_IO_A2F[690],P2F_GPIO114_IN,
Right,19,24,1,gfpga_pad_IO_F2A[689],F2P_GPIO114_OUT,
Right,19,24,0,gfpga_pad_IO_F2A[688],F2P_GPIO114_OE,
Right,18,24,6,gfpga_pad_IO_A2F[678],P2F_GPIO115_IN,
Right,18,24,5,gfpga_pad_IO_F2A[677],F2P_GPIO115_OUT,
Right,18,24,4,gfpga_pad_IO_F2A[676],F2P_GPIO115_OE,
Right,18,24,2,gfpga_pad_IO_A2F[674],P2F_GPIO116_IN,
Right,18,24,1,gfpga_pad_IO_F2A[673],F2P_GPIO116_OUT,
Right,18,24,0,gfpga_pad_IO_F2A[672],F2P_GPIO116_OE,
Right,17,24,14,gfpga_pad_IO_A2F[670],RFU,
Right,17,24,13,gfpga_pad_IO_F2A[669],RFU,
Right,17,24,12,gfpga_pad_IO_F2A[668],RFU,
Right,17,24,10,gfpga_pad_IO_A2F[666],RFU,
Right,17,24,9,gfpga_pad_IO_F2A[665],RFU,
Right,17,24,8,gfpga_pad_IO_F2A[664],RFU,
Right,17,24,6,gfpga_pad_IO_A2F[662],P2F_GPIO117_IN,
Right,17,24,5,gfpga_pad_IO_F2A[661],F2P_GPIO117_OUT,
Right,17,24,4,gfpga_pad_IO_F2A[660],F2P_GPIO117_OE,
Right,17,24,2,gfpga_pad_IO_A2F[658],P2F_GPIO118_IN,

```

(continues on next page)

(continued from previous page)

```

Right,17,24,1,gfpga_pad_IO_F2A[657],F2P_GPIO118_OUT,
Right,17,24,0,gfpga_pad_IO_F2A[656],F2P_GPIO118_OE,
Right,16,24,14,gfpga_pad_IO_A2F[654],RFU,
Right,16,24,13,gfpga_pad_IO_F2A[653],RFU,
Right,16,24,12,gfpga_pad_IO_F2A[652],RFU,
Right,16,24,10,gfpga_pad_IO_A2F[650],RFU,
Right,16,24,9,gfpga_pad_IO_F2A[649],RFU,
Right,16,24,8,gfpga_pad_IO_F2A[648],RFU,
Right,16,24,6,gfpga_pad_IO_A2F[646],P2F_GPIO119_IN,
Right,16,24,5,gfpga_pad_IO_F2A[645],F2P_GPIO119_OUT,
Right,16,24,4,gfpga_pad_IO_F2A[644],F2P_GPIO119_OE,
Right,16,24,2,gfpga_pad_IO_A2F[642],P2F_GPIO120_IN,
Right,16,24,1,gfpga_pad_IO_F2A[641],F2P_GPIO120_OUT,
Right,16,24,0,gfpga_pad_IO_F2A[640],F2P_GPIO120_OE,
Right,15,24,14,gfpga_pad_IO_A2F[638],P2F_GPIO121_IN,
Right,15,24,13,gfpga_pad_IO_F2A[637],F2P_GPIO121_OUT,
Right,15,24,12,gfpga_pad_IO_F2A[636],F2P_GPIO121_OE,
Right,15,24,10,gfpga_pad_IO_A2F[634],P2F_GPIO122_IN,
Right,15,24,9,gfpga_pad_IO_F2A[633],F2P_GPIO122_OUT,
Right,15,24,8,gfpga_pad_IO_F2A[632],F2P_GPIO122_OE,
Right,15,24,6,gfpga_pad_IO_A2F[630],P2F_GPIO123_IN,
Right,15,24,5,gfpga_pad_IO_F2A[629],F2P_GPIO123_OUT,
Right,15,24,4,gfpga_pad_IO_F2A[628],F2P_GPIO123_OE,
Right,15,24,2,gfpga_pad_IO_A2F[626],P2F_GPIO124_IN,
Right,15,24,1,gfpga_pad_IO_F2A[625],F2P_GPIO124_OUT,
Right,15,24,0,gfpga_pad_IO_F2A[624],F2P_GPIO124_OE,
Right,14,24,14,gfpga_pad_IO_A2F[622],P2F_GPIO125_IN,
Right,14,24,13,gfpga_pad_IO_F2A[621],F2P_GPIO125_OUT,
Right,14,24,12,gfpga_pad_IO_F2A[620],F2P_GPIO125_OE,
Right,14,24,10,gfpga_pad_IO_A2F[618],P2F_GPIO126_IN,
Right,14,24,9,gfpga_pad_IO_F2A[617],F2P_GPIO126_OUT,
Right,14,24,8,gfpga_pad_IO_F2A[616],F2P_GPIO126_OE,
Right,14,24,6,gfpga_pad_IO_A2F[614],P2F_GPIO127_IN,
Right,14,24,5,gfpga_pad_IO_F2A[613],F2P_GPIO127_OUT,
Right,14,24,4,gfpga_pad_IO_F2A[612],F2P_GPIO127_OE,
Right,14,24,2,gfpga_pad_IO_A2F[610],P2F_GPIO128_IN,
Right,14,24,1,gfpga_pad_IO_F2A[609],F2P_GPIO128_OUT,
Right,14,24,0,gfpga_pad_IO_F2A[608],F2P_GPIO128_OE,
Right,13,24,14,gfpga_pad_IO_A2F[606],P2F_GPIO129_IN,
Right,13,24,13,gfpga_pad_IO_F2A[605],F2P_GPIO129_OUT,
Right,13,24,12,gfpga_pad_IO_F2A[604],F2P_GPIO129_OE,
Right,13,24,10,gfpga_pad_IO_A2F[602],P2F_GPIO130_IN,
Right,13,24,9,gfpga_pad_IO_F2A[601],F2P_GPIO130_OUT,
Right,13,24,8,gfpga_pad_IO_F2A[600],F2P_GPIO130_OE,
Right,13,24,6,gfpga_pad_IO_A2F[598],P2F_GPIO131_IN,
Right,13,24,5,gfpga_pad_IO_F2A[597],F2P_GPIO131_OUT,
Right,13,24,4,gfpga_pad_IO_F2A[596],F2P_GPIO131_OE,
Right,13,24,2,gfpga_pad_IO_A2F[594],P2F_GPIO132_IN,
Right,13,24,1,gfpga_pad_IO_F2A[593],F2P_GPIO132_OUT,
Right,13,24,0,gfpga_pad_IO_F2A[592],F2P_GPIO132_OE,
Right,12,24,14,gfpga_pad_IO_A2F[590],P2F_GPIO133_IN,
Right,12,24,13,gfpga_pad_IO_F2A[589],F2P_GPIO133_OUT,

```

(continues on next page)

(continued from previous page)

```

Right,12,24,12,gfpga_pad_IO_F2A[588],F2P_GPIO133_OE,
Right,12,24,10,gfpga_pad_IO_A2F[586],P2F_GPIO134_IN,
Right,12,24,9,gfpga_pad_IO_F2A[585],F2P_GPIO134_OUT,
Right,12,24,8,gfpga_pad_IO_F2A[584],F2P_GPIO134_OE,
Right,12,24,6,gfpga_pad_IO_A2F[582],P2F_AES0_RX_IN,
Right,12,24,5,gfpga_pad_IO_F2A[581],F2P_AES0_RX_OUT,
Right,12,24,4,gfpga_pad_IO_F2A[580],F2P_AES0_RX_OE,
Right,12,24,2,gfpga_pad_IO_A2F[578],P2F_AES0_TX_IN,
Right,12,24,1,gfpga_pad_IO_F2A[577],F2P_AES0_TX_OUT,
Right,12,24,0,gfpga_pad_IO_F2A[576],F2P_AES0_TX_OE,
Right,11,24,14,gfpga_pad_IO_A2F[574],P2F_AES1_RX_IN,
Right,11,24,13,gfpga_pad_IO_F2A[573],F2P_AES1_RX_OUT,
Right,11,24,12,gfpga_pad_IO_F2A[572],F2P_AES1_RX_OE,
Right,11,24,10,gfpga_pad_IO_A2F[570],P2F_AES1_TX_IN,
Right,11,24,9,gfpga_pad_IO_F2A[569],F2P_AES1_TX_OUT,
Right,11,24,8,gfpga_pad_IO_F2A[568],F2P_AES1_TX_OE,
Right,11,24,6,gfpga_pad_IO_A2F[566],P2F_AES2_RX_IN,
Right,11,24,5,gfpga_pad_IO_F2A[565],F2P_AES2_RX_OUT,
Right,11,24,4,gfpga_pad_IO_F2A[564],F2P_AES2_RX_OE,
Right,11,24,2,gfpga_pad_IO_A2F[562],P2F_AES2_TX_IN,
Right,11,24,1,gfpga_pad_IO_F2A[561],F2P_AES2_TX_OUT,
Right,11,24,0,gfpga_pad_IO_F2A[560],F2P_AES2_TX_OE,
Right,10,24,14,gfpga_pad_IO_A2F[558],P2F_AES3_RX_IN,
Right,10,24,13,gfpga_pad_IO_F2A[557],F2P_AES3_RX_OUT,
Right,10,24,12,gfpga_pad_IO_F2A[556],F2P_AES3_RX_OE,
Right,10,24,10,gfpga_pad_IO_A2F[554],P2F_AES3_TX_IN,
Right,10,24,9,gfpga_pad_IO_F2A[553],F2P_AES3_TX_OUT,
Right,10,24,8,gfpga_pad_IO_F2A[552],F2P_AES3_TX_OE,
Right,10,24,6,gfpga_pad_IO_A2F[550],P2F_I2C_1_SDA_IN,
Right,10,24,5,gfpga_pad_IO_F2A[549],F2P_I2C_1_SDA_OUT,
Right,10,24,4,gfpga_pad_IO_F2A[548],F2P_I2C_1_SDA_OE,
Right,10,24,2,gfpga_pad_IO_A2F[546],P2F_I2C_1_SCL_IN,
Right,10,24,1,gfpga_pad_IO_F2A[545],F2P_I2C_1_SCL_OUT,
Right,10,24,0,gfpga_pad_IO_F2A[544],F2P_I2C_1_SCL_OE,
Right,9,24,14,gfpga_pad_IO_A2F[542],P2F_WCLKO_IN,
Right,9,24,13,gfpga_pad_IO_F2A[541],F2P_WCLKO_OUT,
Right,9,24,12,gfpga_pad_IO_F2A[540],F2P_WCLKO_OE,
Right,9,24,10,gfpga_pad_IO_A2F[538],P2F_WCLKI_IN,
Right,9,24,9,gfpga_pad_IO_F2A[537],F2P_WCLKI_OUT,
Right,9,24,8,gfpga_pad_IO_F2A[536],F2P_WCLKI_OE,
Right,9,24,6,gfpga_pad_IO_A2F[534],P2F_I2C_G_SDA_IN,
Right,9,24,5,gfpga_pad_IO_F2A[533],F2P_I2C_G_SDA_OUT,
Right,9,24,4,gfpga_pad_IO_F2A[532],F2P_I2C_G_SDA_OE,
Right,9,24,2,gfpga_pad_IO_A2F[530],P2F_I2C_G_SCL_IN,
Right,9,24,1,gfpga_pad_IO_F2A[529],F2P_I2C_G_SCL_OUT,
Right,9,24,0,gfpga_pad_IO_F2A[528],F2P_I2C_G_SCL_OE,
Right,8,24,14,gfpga_pad_IO_A2F[526],P2F_SPI_M_MISO_IN,
Right,8,24,13,gfpga_pad_IO_F2A[525],F2P_SPI_M_MISO_OUT,
Right,8,24,12,gfpga_pad_IO_F2A[524],F2P_SPI_M_MISO_OE,
Right,8,24,10,gfpga_pad_IO_A2F[522],P2F_SPI_M_MOSI_IN,
Right,8,24,9,gfpga_pad_IO_F2A[521],F2P_SPI_M_MOSI_OUT,
Right,8,24,8,gfpga_pad_IO_F2A[520],F2P_SPI_M_MOSI_OE,

```

(continues on next page)

(continued from previous page)

```

Right,8,24,6,gfpga_pad_IO_A2F[518],P2F_SPI_M_CLK_IN,
Right,8,24,5,gfpga_pad_IO_F2A[517],F2P_SPI_M_CLK_OUT,
Right,8,24,4,gfpga_pad_IO_F2A[516],F2P_SPI_M_CLK_OE,
Right,8,24,2,gfpga_pad_IO_A2F[514],P2F_SPI_M_CS_B_IN,
Right,8,24,1,gfpga_pad_IO_F2A[513],F2P_SPI_M_CS_B_OUT,
Right,8,24,0,gfpga_pad_IO_F2A[512],F2P_SPI_M_CS_B_OE,
Right,7,24,14,gfpga_pad_IO_A2F[510],RFU,
Right,7,24,13,gfpga_pad_IO_F2A[509],RFU,
Right,7,24,12,gfpga_pad_IO_F2A[508],RFU,
Right,7,24,10,gfpga_pad_IO_A2F[506],P2F_SPI_S_CS_B_IN,
Right,7,24,9,gfpga_pad_IO_F2A[505],F2P_SPI_S_CS_B_OUT,
Right,7,24,8,gfpga_pad_IO_F2A[504],F2P_SPI_S_CS_B_OE,
Right,7,24,6,gfpga_pad_IO_A2F[502],P2F_SPI_S_MISO_IN,
Right,7,24,5,gfpga_pad_IO_F2A[501],F2P_SPI_S_MISO_OUT,
Right,7,24,4,gfpga_pad_IO_F2A[500],F2P_SPI_S_MISO_OE,
Right,7,24,2,gfpga_pad_IO_A2F[498],P2F_SPI_S_MOSI_IN,
Right,7,24,1,gfpga_pad_IO_F2A[497],F2P_SPI_S_MOSI_OUT,
Right,7,24,0,gfpga_pad_IO_F2A[496],F2P_SPI_S_MOSI_OE,
Right,6,24,14,gfpga_pad_IO_A2F[494],RFU,
Right,6,24,13,gfpga_pad_IO_F2A[493],RFU,
Right,6,24,12,gfpga_pad_IO_F2A[492],RFU,
Right,6,24,10,gfpga_pad_IO_A2F[490],RFU,
Right,6,24,9,gfpga_pad_IO_A2F[489],RFU,
Right,6,24,8,gfpga_pad_IO_A2F[488],RFU,
Right,6,24,6,gfpga_pad_IO_A2F[486],P2F_SPI_S_CLK_IN,
Right,6,24,5,gfpga_pad_IO_F2A[485],F2P_SPI_S_CLK_OUT,
Right,6,24,4,gfpga_pad_IO_F2A[484],F2P_SPI_S_CLK_OE,
Right,6,24,0,gfpga_pad_IO_F2A[480],F2A_LB_CLK_R,
Right,5,24,6,gfpga_pad_IO_A2F[470],P2F_UART_B_RX_IN,
Right,5,24,5,gfpga_pad_IO_F2A[469],F2P_UART_B_RX_OUT,
Right,5,24,4,gfpga_pad_IO_F2A[468],F2P_UART_B_RX_OE,
Right,5,24,2,gfpga_pad_IO_A2F[466],P2F_UART_B_TX_IN,
Right,5,24,1,gfpga_pad_IO_F2A[465],F2P_UART_B_TX_OUT,
Right,5,24,0,gfpga_pad_IO_F2A[464],F2P_UART_B_TX_OE,
Right,4,24,6,gfpga_pad_IO_A2F[454],P2F_MIDI_RX_IN,
Right,4,24,5,gfpga_pad_IO_F2A[453],F2P_MIDI_RX_OUT,
Right,4,24,4,gfpga_pad_IO_F2A[452],F2P_MIDI_RX_OE,
Right,4,24,2,gfpga_pad_IO_A2F[450],P2F_MIDI_TX_IN,
Right,4,24,1,gfpga_pad_IO_F2A[449],F2P_MIDI_TX_OUT,
Right,4,24,0,gfpga_pad_IO_F2A[448],F2P_MIDI_TX_OE,
Bottom,24,21,6,gfpga_pad_IO_A2F[1110],RFU,
Bottom,24,21,5,gfpga_pad_IO_F2A[1109],RFU,
Bottom,24,21,4,gfpga_pad_IO_F2A[1108],RFU,
Bottom,24,21,2,gfpga_pad_IO_A2F[1106],RFU,
Bottom,24,21,1,gfpga_pad_IO_F2A[1105],RFU,
Bottom,24,21,0,gfpga_pad_IO_F2A[1104],RFU,
Bottom,24,20,6,gfpga_pad_IO_A2F[1094],RFU,
Bottom,24,20,5,gfpga_pad_IO_F2A[1093],RFU,
Bottom,24,20,4,gfpga_pad_IO_F2A[1092],RFU,
Bottom,24,20,2,gfpga_pad_IO_A2F[1090],RFU,
Bottom,24,20,1,gfpga_pad_IO_F2A[1089],RFU,
Bottom,24,20,0,gfpga_pad_IO_F2A[1088],RFU,

```

(continues on next page)

(continued from previous page)

```

Bottom,24,19,14,gfpga_pad_IO_A2F[1086],P2F_EEC_IN,
Bottom,24,19,13,gfpga_pad_IO_F2A[1085],F2P_EEC_OUT,
Bottom,24,19,12,gfpga_pad_IO_F2A[1084],F2P_EEC_OE,
Bottom,24,19,10,gfpga_pad_IO_A2F[1082],P2F_GPIO111_IN,
Bottom,24,19,9,gfpga_pad_IO_F2A[1081],F2P_GPIO111_OUT,
Bottom,24,19,8,gfpga_pad_IO_F2A[1080],F2P_GPIO111_OE,
Bottom,24,19,6,gfpga_pad_IO_A2F[1078],P2F_IXS0_SDI_IN,
Bottom,24,19,5,gfpga_pad_IO_F2A[1077],F2P_IXS0_SDI_OUT,
Bottom,24,19,4,gfpga_pad_IO_F2A[1076],F2P_IXS0_SDI_OE,
Bottom,24,19,2,gfpga_pad_IO_A2F[1074],P2F_IXS0_SDO_IN,
Bottom,24,19,1,gfpga_pad_IO_F2A[1073],F2P_IXS0_SDO_OUT,
Bottom,24,19,0,gfpga_pad_IO_F2A[1072],F2P_IXS0_SDO_OE,
Bottom,24,18,14,gfpga_pad_IO_A2F[1070],P2F_IXS1_SDI_IN,
Bottom,24,18,13,gfpga_pad_IO_F2A[1069],F2P_IXS1_SDI_OUT,
Bottom,24,18,12,gfpga_pad_IO_F2A[1068],F2P_IXS1_SDI_OE,
Bottom,24,18,10,gfpga_pad_IO_A2F[1066],P2F_IXS1_SDO_IN,
Bottom,24,18,9,gfpga_pad_IO_F2A[1065],F2P_IXS1_SDO_OUT,
Bottom,24,18,8,gfpga_pad_IO_F2A[1064],F2P_IXS1_SDO_OE,
Bottom,24,18,6,gfpga_pad_IO_A2F[1062],P2F_IXS2_SDI_IN,
Bottom,24,18,5,gfpga_pad_IO_F2A[1061],F2P_IXS2_SDI_OUT,
Bottom,24,18,4,gfpga_pad_IO_F2A[1060],F2P_IXS2_SDI_OE,
Bottom,24,18,2,gfpga_pad_IO_A2F[1058],P2F_IXS2_SDO_IN,
Bottom,24,18,1,gfpga_pad_IO_F2A[1057],F2P_IXS2_SDO_OUT,
Bottom,24,18,0,gfpga_pad_IO_F2A[1056],F2P_IXS2_SDO_OE,
Bottom,24,17,14,gfpga_pad_IO_A2F[1054],P2F_IXS3_SDI_IN,
Bottom,24,17,13,gfpga_pad_IO_F2A[1053],F2P_IXS3_SDI_OUT,
Bottom,24,17,12,gfpga_pad_IO_F2A[1052],F2P_IXS3_SDI_OE,
Bottom,24,17,10,gfpga_pad_IO_A2F[1050],P2F_IXS3_SDO_IN,
Bottom,24,17,9,gfpga_pad_IO_F2A[1049],F2P_IXS3_SDO_OUT,
Bottom,24,17,8,gfpga_pad_IO_F2A[1048],F2P_IXS3_SDO_OE,
Bottom,24,17,6,gfpga_pad_IO_A2F[1046],P2F_IXS4_SDI_IN,
Bottom,24,17,5,gfpga_pad_IO_F2A[1045],F2P_IXS4_SDI_OUT,
Bottom,24,17,4,gfpga_pad_IO_F2A[1044],F2P_IXS4_SDI_OE,
Bottom,24,17,2,gfpga_pad_IO_A2F[1042],P2F_IXS4_SDO_IN,
Bottom,24,17,1,gfpga_pad_IO_F2A[1041],F2P_IXS4_SDO_OUT,
Bottom,24,17,0,gfpga_pad_IO_F2A[1040],F2P_IXS4_SDO_OE,
Bottom,24,16,14,gfpga_pad_IO_A2F[1038],P2F_IXS5_SDI_IN,
Bottom,24,16,13,gfpga_pad_IO_F2A[1037],F2P_IXS5_SDI_OUT,
Bottom,24,16,12,gfpga_pad_IO_F2A[1036],F2P_IXS5_SDI_OE,
Bottom,24,16,10,gfpga_pad_IO_A2F[1034],P2F_IXS5_SDO_IN,
Bottom,24,16,9,gfpga_pad_IO_F2A[1033],F2P_IXS5_SDO_OUT,
Bottom,24,16,8,gfpga_pad_IO_F2A[1032],F2P_IXS5_SDO_OE,
Bottom,24,16,6,gfpga_pad_IO_A2F[1030],P2F_IXS6_SDI_IN,
Bottom,24,16,5,gfpga_pad_IO_F2A[1029],F2P_IXS6_SDI_OUT,
Bottom,24,16,4,gfpga_pad_IO_F2A[1028],F2P_IXS6_SDI_OE,
Bottom,24,16,2,gfpga_pad_IO_A2F[1026],P2F_IXS6_SDO_IN,
Bottom,24,16,1,gfpga_pad_IO_F2A[1025],F2P_IXS6_SDO_OUT,
Bottom,24,16,0,gfpga_pad_IO_F2A[1024],F2P_IXS6_SDO_OE,
Bottom,24,15,14,gfpga_pad_IO_A2F[1022],P2F_IXS7_SDI_IN,
Bottom,24,15,13,gfpga_pad_IO_F2A[1021],F2P_IXS7_SDI_OUT,
Bottom,24,15,12,gfpga_pad_IO_F2A[1020],F2P_IXS7_SDI_OE,
Bottom,24,15,10,gfpga_pad_IO_A2F[1018],P2F_IXS7_SDO_IN,

```

(continues on next page)



(continued from previous page)

```

Bottom,24,15,9,gfpga_pad_IO_F2A[1017],F2P_IXS7_SDO_OUT,
Bottom,24,15,8,gfpga_pad_IO_F2A[1016],F2P_IXS7_SDO_OE,
Bottom,24,15,6,gfpga_pad_IO_A2F[1014],P2F_FCLK0_0_IN,
Bottom,24,15,5,gfpga_pad_IO_F2A[1013],F2P_FCLK0_0_OUT,
Bottom,24,15,4,gfpga_pad_IO_F2A[1012],F2P_FCLK0_0_OE,
Bottom,24,15,2,gfpga_pad_IO_A2F[1010],P2F_FCLK0_1_IN,
Bottom,24,15,1,gfpga_pad_IO_F2A[1009],F2P_FCLK0_1_OUT,
Bottom,24,15,0,gfpga_pad_IO_F2A[1008],F2P_FCLK0_1_OE,
Bottom,24,14,14,gfpga_pad_IO_A2F[1006],P2F_BCLK0_0_IN,
Bottom,24,14,13,gfpga_pad_IO_F2A[1005],F2P_BCLK0_0_OUT,
Bottom,24,14,12,gfpga_pad_IO_F2A[1004],F2P_BCLK0_0_OE,
Bottom,24,14,10,gfpga_pad_IO_A2F[1002],P2F_BCLK0_1_IN,
Bottom,24,14,9,gfpga_pad_IO_F2A[1001],F2P_BCLK0_1_OUT,
Bottom,24,14,8,gfpga_pad_IO_F2A[1000],F2P_BCLK0_1_OE,
Bottom,24,14,6,gfpga_pad_IO_A2F[998],P2F_BCLK0_2_IN,
Bottom,24,14,5,gfpga_pad_IO_F2A[997],F2P_BCLK0_2_OUT,
Bottom,24,14,4,gfpga_pad_IO_F2A[996],F2P_BCLK0_2_OE,
Bottom,24,14,2,gfpga_pad_IO_A2F[994],P2F_BCLKIO_IN,
Bottom,24,14,1,gfpga_pad_IO_F2A[993],F2P_BCLKIO_OUT,
Bottom,24,14,0,gfpga_pad_IO_F2A[992],F2P_BCLKIO_OE,
Bottom,24,13,14,gfpga_pad_IO_A2F[990],P2F_MCLKIO_IN,
Bottom,24,13,13,gfpga_pad_IO_F2A[989],F2P_MCLKIO_OUT,
Bottom,24,13,12,gfpga_pad_IO_F2A[988],F2P_MCLKIO_OE,
Bottom,24,13,10,gfpga_pad_IO_A2F[986],P2F_MCLK0_0_IN,
Bottom,24,13,9,gfpga_pad_IO_F2A[985],F2P_MCLK0_0_OUT,
Bottom,24,13,8,gfpga_pad_IO_F2A[984],F2P_MCLK0_0_OE,
Bottom,24,13,6,gfpga_pad_IO_A2F[982],P2F_MCLK0_1_IN,
Bottom,24,13,5,gfpga_pad_IO_F2A[981],F2P_MCLK0_1_OUT,
Bottom,24,13,4,gfpga_pad_IO_F2A[980],F2P_MCLK0_1_OE,
Bottom,24,13,2,gfpga_pad_IO_A2F[978],P2F_MCLK0_2_IN,
Bottom,24,13,1,gfpga_pad_IO_F2A[977],F2P_MCLK0_2_OUT,
Bottom,24,13,0,gfpga_pad_IO_F2A[976],F2P_MCLK0_2_OE,
Bottom,24,12,0,gfpga_pad_IO_A2F[960],P2F_FCLKIO_HS,
Bottom,24,11,7,gfpga_pad_IO_A2F[951],P2F_IXS8_SDI_HS,
Bottom,24,11,6,gfpga_pad_IO_A2F[950],P2F_IXS9_SDI_HS,
Bottom,24,11,5,gfpga_pad_IO_A2F[949],P2F_IXS10_SDI_HS,
Bottom,24,11,4,gfpga_pad_IO_A2F[948],P2F_IXS11_SDI_HS,
Bottom,24,11,3,gfpga_pad_IO_A2F[947],P2F_IXS12_SDI_HS,
Bottom,24,11,2,gfpga_pad_IO_A2F[946],P2F_IXS13_SDI_HS,
Bottom,24,11,1,gfpga_pad_IO_A2F[945],P2F_IXS14_SDI_HS,
Bottom,24,11,0,gfpga_pad_IO_A2F[944],P2F_IXS15_SDI_HS,
Bottom,24,10,4,gfpga_pad_IO_F2A[932],F2P_FCLK0_0_HS,
Bottom,24,10,3,gfpga_pad_IO_F2A[931],F2P_FCLK0_1_HS,
Bottom,24,10,2,gfpga_pad_IO_F2A[930],F2P_FCLKIO_HS,
Bottom,24,10,1,gfpga_pad_IO_F2A[929],F2P_FCLK0_2_HS,
Bottom,24,10,0,gfpga_pad_IO_F2A[928],F2P_FCLK0_3_HS,
Bottom,24,9,7,gfpga_pad_IO_F2A[919],F2P_IXS8_SDO_HS,
Bottom,24,9,6,gfpga_pad_IO_F2A[918],F2P_IXS9_SDO_HS,
Bottom,24,9,5,gfpga_pad_IO_F2A[917],F2P_IXS10_SDO_HS,
Bottom,24,9,4,gfpga_pad_IO_F2A[916],F2P_IXS11_SDO_HS,
Bottom,24,9,3,gfpga_pad_IO_F2A[915],F2P_IXS12_SDO_HS,
Bottom,24,9,2,gfpga_pad_IO_F2A[914],F2P_IXS13_SDO_HS,

```

(continues on next page)

(continued from previous page)

```

Bottom,24,9,1,gfpga_pad_IO_F2A[913],F2P_IXS14_SDO_HS,
Bottom,24,9,0,gfpga_pad_IO_F2A[912],F2P_IXS15_SDO_HS,
Bottom,24,8,14,gfpga_pad_IO_A2F[910],P2F_FCLKIO_IN,
Bottom,24,8,13,gfpga_pad_IO_F2A[909],F2P_FCLKIO_OUT,
Bottom,24,8,12,gfpga_pad_IO_F2A[908],F2P_FCLKIO_OE,
Bottom,24,8,10,gfpga_pad_IO_A2F[906],P2F_FCLK0_2_IN,
Bottom,24,8,9,gfpga_pad_IO_F2A[905],F2P_FCLK0_2_OUT,
Bottom,24,8,8,gfpga_pad_IO_F2A[904],F2P_FCLK0_2_OE,
Bottom,24,8,6,gfpga_pad_IO_A2F[902],P2F_FCLK0_3_IN,
Bottom,24,8,5,gfpga_pad_IO_F2A[901],F2P_FCLK0_3_OUT,
Bottom,24,8,4,gfpga_pad_IO_F2A[900],F2P_FCLK0_3_OE,
Bottom,24,8,1,gfpga_pad_IO_F2A[897],F2A_LB_CLK_B,
Bottom,24,7,14,gfpga_pad_IO_A2F[894],P2F_IXS8_SDI_IN,
Bottom,24,7,13,gfpga_pad_IO_F2A[893],F2P_IXS8_SDI_OUT,
Bottom,24,7,12,gfpga_pad_IO_F2A[892],F2P_IXS8_SDI_OE,
Bottom,24,7,10,gfpga_pad_IO_A2F[890],P2F_IXS8_SDO_IN,
Bottom,24,7,9,gfpga_pad_IO_F2A[889],F2P_IXS8_SDO_OUT,
Bottom,24,7,8,gfpga_pad_IO_F2A[888],F2P_IXS8_SDO_OE,
Bottom,24,7,6,gfpga_pad_IO_A2F[886],P2F_IXS9_SDI_IN,
Bottom,24,7,5,gfpga_pad_IO_F2A[885],F2P_IXS9_SDI_OUT,
Bottom,24,7,4,gfpga_pad_IO_F2A[884],F2P_IXS9_SDI_OE,
Bottom,24,7,2,gfpga_pad_IO_A2F[882],P2F_IXS9_SDO_IN,
Bottom,24,7,1,gfpga_pad_IO_F2A[881],F2P_IXS9_SDO_OUT,
Bottom,24,7,0,gfpga_pad_IO_F2A[880],F2P_IXS9_SDO_OE,
Bottom,24,6,14,gfpga_pad_IO_A2F[878],P2F_IXS10_SDI_IN,
Bottom,24,6,13,gfpga_pad_IO_F2A[877],F2P_IXS10_SDI_OUT,
Bottom,24,6,12,gfpga_pad_IO_F2A[876],F2P_IXS10_SDI_OE,
Bottom,24,6,10,gfpga_pad_IO_A2F[874],P2F_IXS10_SDO_IN,
Bottom,24,6,9,gfpga_pad_IO_F2A[873],F2P_IXS10_SDO_OUT,
Bottom,24,6,8,gfpga_pad_IO_F2A[872],F2P_IXS10_SDO_OE,
Bottom,24,6,6,gfpga_pad_IO_A2F[870],P2F_IXS11_SDI_IN,
Bottom,24,6,5,gfpga_pad_IO_F2A[869],F2P_IXS11_SDI_OUT,
Bottom,24,6,4,gfpga_pad_IO_F2A[868],F2P_IXS11_SDI_OE,
Bottom,24,6,2,gfpga_pad_IO_A2F[866],P2F_IXS11_SDO_IN,
Bottom,24,6,1,gfpga_pad_IO_F2A[865],F2P_IXS11_SDO_OUT,
Bottom,24,6,0,gfpga_pad_IO_F2A[864],F2P_IXS11_SDO_OE,
Bottom,24,5,14,gfpga_pad_IO_A2F[862],P2F_IXS12_SDI_IN,
Bottom,24,5,13,gfpga_pad_IO_F2A[861],F2P_IXS12_SDI_OUT,
Bottom,24,5,12,gfpga_pad_IO_F2A[860],F2P_IXS12_SDI_OE,
Bottom,24,5,10,gfpga_pad_IO_A2F[858],P2F_IXS12_SDO_IN,
Bottom,24,5,9,gfpga_pad_IO_F2A[857],F2P_IXS12_SDO_OUT,
Bottom,24,5,8,gfpga_pad_IO_F2A[856],F2P_IXS12_SDO_OE,
Bottom,24,5,6,gfpga_pad_IO_A2F[854],P2F_IXS13_SDI_IN,
Bottom,24,5,5,gfpga_pad_IO_F2A[853],F2P_IXS13_SDI_OUT,
Bottom,24,5,4,gfpga_pad_IO_F2A[852],F2P_IXS13_SDI_OE,
Bottom,24,5,2,gfpga_pad_IO_A2F[850],P2F_IXS13_SDO_IN,
Bottom,24,5,1,gfpga_pad_IO_F2A[849],F2P_IXS13_SDO_OUT,
Bottom,24,5,0,gfpga_pad_IO_F2A[848],F2P_IXS13_SDO_OE,
Bottom,24,4,14,gfpga_pad_IO_A2F[846],P2F_IXS14_SDI_IN,
Bottom,24,4,13,gfpga_pad_IO_F2A[845],F2P_IXS14_SDI_OUT,
Bottom,24,4,12,gfpga_pad_IO_F2A[844],F2P_IXS14_SDI_OE,
Bottom,24,4,10,gfpga_pad_IO_A2F[842],P2F_IXS14_SDO_IN,

```

(continues on next page)

(continued from previous page)

```

Bottom,24,4,9,gfpga_pad_IO_F2A[841],F2P_IXS14_SDO_OUT,
Bottom,24,4,8,gfpga_pad_IO_F2A[840],F2P_IXS14_SDO_OE,
Bottom,24,4,6,gfpga_pad_IO_A2F[838],P2F_IXS15_SDI_IN,
Bottom,24,4,5,gfpga_pad_IO_F2A[837],F2P_IXS15_SDI_OUT,
Bottom,24,4,4,gfpga_pad_IO_F2A[836],F2P_IXS15_SDI_OE,
Bottom,24,4,2,gfpga_pad_IO_A2F[834],P2F_IXS15_SDO_IN,
Bottom,24,4,1,gfpga_pad_IO_F2A[833],F2P_IXS15_SDO_OUT,
Bottom,24,4,0,gfpga_pad_IO_F2A[832],F2P_IXS15_SDO_OE,
Bottom,24,3,6,gfpga_pad_IO_A2F[822],RFU,
Bottom,24,3,5,gfpga_pad_IO_F2A[821],RFU,
Bottom,24,3,4,gfpga_pad_IO_F2A[820],RFU,
Bottom,24,3,2,gfpga_pad_IO_A2F[818],RFU,
Bottom,24,3,1,gfpga_pad_IO_F2A[817],RFU,
Bottom,24,3,0,gfpga_pad_IO_F2A[816],RFU,
Bottom,24,2,6,gfpga_pad_IO_A2F[806],RFU,
Bottom,24,2,5,gfpga_pad_IO_F2A[805],RFU,
Bottom,24,2,4,gfpga_pad_IO_F2A[804],RFU,
Bottom,24,2,2,gfpga_pad_IO_A2F[802],RFU,
Bottom,24,2,1,gfpga_pad_IO_F2A[801],RFU,
Bottom,24,2,0,gfpga_pad_IO_F2A[800],RFU,
Left,23,0,15,gfpga_pad_IO_F2A[1535],F2A_GPI06_31,
Left,23,0,14,gfpga_pad_IO_F2A[1534],F2A_GPI06_30,
Left,23,0,13,gfpga_pad_IO_F2A[1533],F2A_GPI06_29,
Left,23,0,12,gfpga_pad_IO_F2A[1532],F2A_GPI06_28,
Left,23,0,11,gfpga_pad_IO_A2F[1531],A2F_GPI06_31,
Left,23,0,10,gfpga_pad_IO_A2F[1530],A2F_GPI06_30,
Left,23,0,9,gfpga_pad_IO_A2F[1529],A2F_GPI06_29,
Left,23,0,8,gfpga_pad_IO_A2F[1528],A2F_GPI06_28,
Left,22,0,15,gfpga_pad_IO_F2A[1519],F2A_GPI06_27,
Left,22,0,14,gfpga_pad_IO_F2A[1518],F2A_GPI06_26,
Left,22,0,13,gfpga_pad_IO_F2A[1517],F2A_GPI06_25,
Left,22,0,12,gfpga_pad_IO_F2A[1516],F2A_GPI06_24,
Left,22,0,11,gfpga_pad_IO_A2F[1515],A2F_GPI06_27,
Left,22,0,10,gfpga_pad_IO_A2F[1514],A2F_GPI06_26,
Left,22,0,9,gfpga_pad_IO_A2F[1513],A2F_GPI06_25,
Left,22,0,8,gfpga_pad_IO_A2F[1512],A2F_GPI06_24,
Left,21,0,15,gfpga_pad_IO_F2A[1503],F2A_GPI06_23,
Left,21,0,14,gfpga_pad_IO_F2A[1502],F2A_GPI06_22,
Left,21,0,13,gfpga_pad_IO_F2A[1501],F2A_GPI06_21,
Left,21,0,12,gfpga_pad_IO_F2A[1500],F2A_GPI06_20,
Left,21,0,11,gfpga_pad_IO_A2F[1499],A2F_GPI06_23,
Left,21,0,10,gfpga_pad_IO_A2F[1498],A2F_GPI06_22,
Left,21,0,9,gfpga_pad_IO_A2F[1497],A2F_GPI06_21,
Left,21,0,8,gfpga_pad_IO_A2F[1496],A2F_GPI06_20,
Left,21,0,0,gfpga_pad_IO_F2A[1488],F2A_LB_CLK_L,
Left,20,0,15,gfpga_pad_IO_F2A[1487],F2A_GPI06_19,
Left,20,0,14,gfpga_pad_IO_F2A[1486],F2A_GPI06_18,
Left,20,0,13,gfpga_pad_IO_F2A[1485],F2A_GPI06_17,
Left,20,0,12,gfpga_pad_IO_F2A[1484],F2A_GPI06_16,
Left,20,0,11,gfpga_pad_IO_A2F[1483],A2F_GPI06_19,
Left,20,0,10,gfpga_pad_IO_A2F[1482],A2F_GPI06_18,
Left,20,0,9,gfpga_pad_IO_A2F[1481],A2F_GPI06_17,

```

(continues on next page)

(continued from previous page)

```

Left,20,0,8,gfpga_pad_IO_A2F[1480],A2F_GPIO6_16,
Left,20,0,1,gfpga_pad_IO_F2A[1473],F2A_ACM_AES1_RX_FCLK,
Left,20,0,0,gfpga_pad_IO_F2A[1472],F2A_ACM_WC_FCLK,
Left,19,0,15,gfpga_pad_IO_F2A[1471],F2A_GPIO6_15,
Left,19,0,14,gfpga_pad_IO_F2A[1470],F2A_GPIO6_14,
Left,19,0,13,gfpga_pad_IO_F2A[1469],F2A_GPIO6_13,
Left,19,0,12,gfpga_pad_IO_F2A[1468],F2A_GPIO6_12,
Left,19,0,11,gfpga_pad_IO_A2F[1467],A2F_GPIO6_15,
Left,19,0,10,gfpga_pad_IO_A2F[1466],A2F_GPIO6_14,
Left,19,0,9,gfpga_pad_IO_A2F[1465],A2F_GPIO6_13,
Left,19,0,8,gfpga_pad_IO_A2F[1464],A2F_GPIO6_12,
Left,19,0,3,gfpga_pad_IO_A2F[1459],RFU,
Left,19,0,1,gfpga_pad_IO_F2A[1457],F2A_ACM_AES0_RX_FCLK,
Left,19,0,0,gfpga_pad_IO_F2A[1456],F2A_ACM_IXS_RX_FCLK,
Left,18,0,15,gfpga_pad_IO_F2A[1455],F2A_GPIO6_11,
Left,18,0,14,gfpga_pad_IO_F2A[1454],F2A_GPIO6_10,
Left,18,0,13,gfpga_pad_IO_F2A[1453],F2A_GPIO6_9,
Left,18,0,12,gfpga_pad_IO_F2A[1452],F2A_GPIO6_8,
Left,18,0,11,gfpga_pad_IO_A2F[1451],A2F_GPIO6_11,
Left,18,0,10,gfpga_pad_IO_A2F[1450],A2F_GPIO6_10,
Left,18,0,9,gfpga_pad_IO_A2F[1449],A2F_GPIO6_9,
Left,18,0,8,gfpga_pad_IO_A2F[1448],A2F_GPIO6_8,
Left,18,0,5,gfpga_pad_IO_A2F[1445],A2F_GPIO7_WR_ACK,
Left,18,0,4,gfpga_pad_IO_A2F[1444],A2F_GPIO7_RD_ACK,
Left,18,0,3,gfpga_pad_IO_A2F[1443],RFU,
Left,18,0,2,gfpga_pad_IO_F2A[1442],F2A_ACM_AES1_RX_BCLK,
Left,18,0,1,gfpga_pad_IO_F2A[1441],F2A_ACM_AES0_RX_BCLK,
Left,18,0,0,gfpga_pad_IO_F2A[1440],F2A_ACM_IXS_RX_BCLK,
Left,17,0,15,gfpga_pad_IO_F2A[1439],F2A_GPIO6_7,
Left,17,0,14,gfpga_pad_IO_F2A[1438],F2A_GPIO6_6,
Left,17,0,13,gfpga_pad_IO_F2A[1437],F2A_GPIO6_5,
Left,17,0,12,gfpga_pad_IO_F2A[1436],F2A_GPIO6_4,
Left,17,0,11,gfpga_pad_IO_A2F[1435],A2F_GPIO6_7,
Left,17,0,10,gfpga_pad_IO_A2F[1434],A2F_GPIO6_6,
Left,17,0,9,gfpga_pad_IO_A2F[1433],A2F_GPIO6_5,
Left,17,0,8,gfpga_pad_IO_A2F[1432],A2F_GPIO6_4,
Left,17,0,7,gfpga_pad_IO_F2A[1431],RFU,
Left,17,0,6,gfpga_pad_IO_F2A[1430],RFU,
Left,17,0,5,gfpga_pad_IO_A2F[1429],A2F_GPIO6_WR_ACK,
Left,17,0,4,gfpga_pad_IO_A2F[1428],A2F_GPIO6_RD_ACK,
Left,17,0,3,gfpga_pad_IO_A2F[1427],RFU,
Left,17,0,1,gfpga_pad_IO_F2A[1425],F2A_ACM_FPGA_FCLK,
Left,17,0,0,gfpga_pad_IO_F2A[1424],F2A_ACM_APLL_FCLK,
Left,16,0,15,gfpga_pad_IO_F2A[1423],F2A_GPIO6_3,
Left,16,0,14,gfpga_pad_IO_F2A[1422],F2A_GPIO6_2,
Left,16,0,13,gfpga_pad_IO_F2A[1421],F2A_GPIO6_1,
Left,16,0,12,gfpga_pad_IO_F2A[1420],F2A_GPIO6_0,
Left,16,0,11,gfpga_pad_IO_A2F[1419],A2F_GPIO6_3,
Left,16,0,10,gfpga_pad_IO_A2F[1418],A2F_GPIO6_2,
Left,16,0,9,gfpga_pad_IO_A2F[1417],A2F_GPIO6_1,
Left,16,0,8,gfpga_pad_IO_A2F[1416],A2F_GPIO6_0,
Left,16,0,7,gfpga_pad_IO_F2A[1415],RFIU,

```

(continues on next page)

(continued from previous page)

```

Left,16,0,6,gfpga_pad_IO_F2A[1414],RFU,
Left,16,0,5,gfpga_pad_IO_A2F[1413],A2F_GPIO4_WR_ACK,
Left,16,0,4,gfpga_pad_IO_A2F[1412],A2F_GPIO4_RD_ACK,
Left,16,0,3,gfpga_pad_IO_A2F[1411],RFU,
Left,16,0,1,gfpga_pad_IO_F2A[1409],F2A_ACM_FPGA_BCLK,
Left,16,0,0,gfpga_pad_IO_F2A[1408],F2A_ACM_APLL_BCLK,
Left,15,0,15,gfpga_pad_IO_F2A[1407],F2A_GPIO5_31,
Left,15,0,14,gfpga_pad_IO_F2A[1406],F2A_GPIO5_30,
Left,15,0,13,gfpga_pad_IO_F2A[1405],F2A_GPIO5_29,
Left,15,0,12,gfpga_pad_IO_F2A[1404],F2A_GPIO5_28,
Left,15,0,11,gfpga_pad_IO_A2F[1403],A2F_GPIO5_31,
Left,15,0,10,gfpga_pad_IO_A2F[1402],A2F_GPIO5_30,
Left,15,0,9,gfpga_pad_IO_A2F[1401],A2F_GPIO5_29,
Left,15,0,8,gfpga_pad_IO_A2F[1400],A2F_GPIO5_28,
Left,15,0,7,gfpga_pad_IO_A2F[1399],RFU,
Left,15,0,6,gfpga_pad_IO_A2F[1398],RFU,
Left,15,0,4,gfpga_pad_IO_A2F[1396],A2F_FIFOW_AEMPTY_HS,
Left,15,0,2,gfpga_pad_IO_F2A[1394],F2A_FIFO_WE_HS,
Left,15,0,1,gfpga_pad_IO_A2F[1393],A2F_FIFOW_FULL_HS,
Left,15,0,0,gfpga_pad_IO_A2F[1392],A2F_FIFOW_EMPTY_HS,
Left,14,0,15,gfpga_pad_IO_F2A[1391],F2A_GPIO5_27,
Left,14,0,14,gfpga_pad_IO_F2A[1390],F2A_GPIO5_26,
Left,14,0,13,gfpga_pad_IO_F2A[1389],F2A_GPIO5_25,
Left,14,0,12,gfpga_pad_IO_F2A[1388],F2A_GPIO5_24,
Left,14,0,11,gfpga_pad_IO_A2F[1387],A2F_GPIO5_27,
Left,14,0,10,gfpga_pad_IO_A2F[1386],A2F_GPIO5_26,
Left,14,0,9,gfpga_pad_IO_A2F[1385],A2F_GPIO5_25,
Left,14,0,8,gfpga_pad_IO_A2F[1384],A2F_GPIO5_24,
Left,14,0,7,gfpga_pad_IO_A2F[1383],A2F_FIFO_WD15_HS,
Left,14,0,6,gfpga_pad_IO_A2F[1382],A2F_FIFO_WD14_HS,
Left,14,0,5,gfpga_pad_IO_A2F[1381],A2F_FIFO_WD13_HS,
Left,14,0,4,gfpga_pad_IO_A2F[1380],A2F_FIFO_WD12_HS,
Left,14,0,3,gfpga_pad_IO_A2F[1379],A2F_FIFO_WD11_HS,
Left,14,0,2,gfpga_pad_IO_A2F[1378],A2F_FIFO_WD10_HS,
Left,14,0,1,gfpga_pad_IO_A2F[1377],A2F_FIFO_WD9_HS,
Left,14,0,0,gfpga_pad_IO_A2F[1376],A2F_FIFO_WD8_HS,
Left,13,0,15,gfpga_pad_IO_F2A[1375],F2A_GPIO5_23,
Left,13,0,14,gfpga_pad_IO_F2A[1374],F2A_GPIO5_22,
Left,13,0,13,gfpga_pad_IO_F2A[1373],F2A_GPIO5_21,
Left,13,0,12,gfpga_pad_IO_F2A[1372],F2A_GPIO5_20,
Left,13,0,11,gfpga_pad_IO_A2F[1371],A2F_GPIO5_23,
Left,13,0,10,gfpga_pad_IO_A2F[1370],A2F_GPIO5_22,
Left,13,0,9,gfpga_pad_IO_A2F[1369],A2F_GPIO5_21,
Left,13,0,8,gfpga_pad_IO_A2F[1368],A2F_GPIO5_20,
Left,13,0,7,gfpga_pad_IO_A2F[1367],A2F_FIFO_WD7_HS,
Left,13,0,6,gfpga_pad_IO_A2F[1366],A2F_FIFO_WD6_HS,
Left,13,0,5,gfpga_pad_IO_A2F[1365],A2F_FIFO_WD5_HS,
Left,13,0,4,gfpga_pad_IO_A2F[1364],A2F_FIFO_WD4_HS,
Left,13,0,3,gfpga_pad_IO_A2F[1363],A2F_FIFO_WD3_HS,
Left,13,0,2,gfpga_pad_IO_A2F[1362],A2F_FIFO_WD2_HS,
Left,13,0,1,gfpga_pad_IO_A2F[1361],A2F_FIFO_WD1_HS,
Left,13,0,0,gfpga_pad_IO_A2F[1360],A2F_FIFO_WD0_HS,

```

(continues on next page)

(continued from previous page)

```

Left,12,0,15,gfpga_pad_IO_F2A[1359],F2A_GPIO5_19,
Left,12,0,14,gfpga_pad_IO_F2A[1358],F2A_GPIO5_18,
Left,12,0,13,gfpga_pad_IO_F2A[1357],F2A_GPIO5_17,
Left,12,0,12,gfpga_pad_IO_F2A[1356],F2A_GPIO5_16,
Left,12,0,11,gfpga_pad_IO_A2F[1355],A2F_GPIO5_19,
Left,12,0,10,gfpga_pad_IO_A2F[1354],A2F_GPIO5_18,
Left,12,0,9,gfpga_pad_IO_A2F[1353],A2F_GPIO5_17,
Left,12,0,8,gfpga_pad_IO_A2F[1352],A2F_GPIO5_16,
Left,12,0,7,gfpga_pad_IO_F2A[1351],F2A_FIFO_RD15_HS,
Left,12,0,6,gfpga_pad_IO_F2A[1350],F2A_FIFO_RD14_HS,
Left,12,0,5,gfpga_pad_IO_F2A[1349],F2A_FIFO_RD13_HS,
Left,12,0,4,gfpga_pad_IO_F2A[1348],F2A_FIFO_RD12_HS,
Left,12,0,3,gfpga_pad_IO_F2A[1347],F2A_FIFO_RD11_HS,
Left,12,0,2,gfpga_pad_IO_F2A[1346],F2A_FIFO_RD10_HS,
Left,12,0,1,gfpga_pad_IO_F2A[1345],F2A_FIFO_RD9_HS,
Left,12,0,0,gfpga_pad_IO_F2A[1344],F2A_FIFO_RD8_HS,
Left,11,0,15,gfpga_pad_IO_F2A[1343],F2A_GPIO5_15,
Left,11,0,14,gfpga_pad_IO_F2A[1342],F2A_GPIO5_14,
Left,11,0,13,gfpga_pad_IO_F2A[1341],F2A_GPIO5_13,
Left,11,0,12,gfpga_pad_IO_F2A[1340],F2A_GPIO5_12,
Left,11,0,11,gfpga_pad_IO_A2F[1339],A2F_GPIO5_15,
Left,11,0,10,gfpga_pad_IO_A2F[1338],A2F_GPIO5_14,
Left,11,0,9,gfpga_pad_IO_A2F[1337],A2F_GPIO5_13,
Left,11,0,8,gfpga_pad_IO_A2F[1336],A2F_GPIO5_12,
Left,11,0,7,gfpga_pad_IO_F2A[1335],F2A_FIFO_RD7_HS,
Left,11,0,6,gfpga_pad_IO_F2A[1334],F2A_FIFO_RD6_HS,
Left,11,0,5,gfpga_pad_IO_F2A[1333],F2A_FIFO_RD5_HS,
Left,11,0,4,gfpga_pad_IO_F2A[1332],F2A_FIFO_RD4_HS,
Left,11,0,3,gfpga_pad_IO_F2A[1331],F2A_FIFO_RD3_HS,
Left,11,0,2,gfpga_pad_IO_F2A[1330],F2A_FIFO_RD2_HS,
Left,11,0,1,gfpga_pad_IO_F2A[1329],F2A_FIFO_RD1_HS,
Left,11,0,0,gfpga_pad_IO_F2A[1328],F2A_FIFO_RD0_HS,
Left,10,0,15,gfpga_pad_IO_F2A[1327],F2A_GPIO5_11,
Left,10,0,14,gfpga_pad_IO_F2A[1326],F2A_GPIO5_10,
Left,10,0,13,gfpga_pad_IO_F2A[1325],F2A_GPIO5_9,
Left,10,0,12,gfpga_pad_IO_F2A[1324],F2A_GPIO5_8,
Left,10,0,11,gfpga_pad_IO_A2F[1323],A2F_GPIO5_11,
Left,10,0,10,gfpga_pad_IO_A2F[1322],A2F_GPIO5_10,
Left,10,0,9,gfpga_pad_IO_A2F[1321],A2F_GPIO5_9,
Left,10,0,8,gfpga_pad_IO_A2F[1320],A2F_GPIO5_8,
Left,10,0,7,gfpga_pad_IO_A2F[1319],RFU,
Left,10,0,6,gfpga_pad_IO_A2F[1318],RFU,
Left,10,0,5,gfpga_pad_IO_A2F[1317],A2F_FIFOR_AFULL_HS,
Left,10,0,2,gfpga_pad_IO_F2A[1314],F2A_FIFO_RE_HS,
Left,10,0,1,gfpga_pad_IO_A2F[1313],A2F_FIFOR_FULL_HS,
Left,10,0,0,gfpga_pad_IO_A2F[1312],A2F_FIFOR_EMPTY_HS,
Left,9,0,15,gfpga_pad_IO_F2A[1311],F2A_GPIO5_7,
Left,9,0,14,gfpga_pad_IO_F2A[1310],F2A_GPIO5_6,
Left,9,0,13,gfpga_pad_IO_F2A[1309],F2A_GPIO5_5,
Left,9,0,12,gfpga_pad_IO_F2A[1308],F2A_GPIO5_4,
Left,9,0,11,gfpga_pad_IO_A2F[1307],A2F_GPIO5_7,
Left,9,0,10,gfpga_pad_IO_A2F[1306],A2F_GPIO5_6,

```

(continues on next page)

(continued from previous page)

```

Left,9,0,9,gfpaga_pad_IO_A2F[1305],A2F_GPIO5_5,
Left,9,0,8,gfpaga_pad_IO_A2F[1304],A2F_GPIO5_4,
Left,9,0,7,gfpaga_pad_IO_A2F[1303],A2F_IXS15_SDO_HS,
Left,9,0,6,gfpaga_pad_IO_A2F[1302],A2F_IXS14_SDO_HS,
Left,9,0,5,gfpaga_pad_IO_A2F[1301],A2F_IXS13_SDO_HS,
Left,9,0,4,gfpaga_pad_IO_A2F[1300],A2F_IXS12_SDO_HS,
Left,9,0,3,gfpaga_pad_IO_A2F[1299],A2F_IXS11_SDO_HS,
Left,9,0,2,gfpaga_pad_IO_A2F[1298],A2F_IXS10_SDO_HS,
Left,9,0,1,gfpaga_pad_IO_A2F[1297],A2F_IXS9_SDO_HS,
Left,9,0,0,gfpaga_pad_IO_A2F[1296],A2F_IXS8_SDO_HS,
Left,8,0,15,gfpaga_pad_IO_F2A[1295],F2A_GPIO5_3,
Left,8,0,14,gfpaga_pad_IO_F2A[1294],F2A_GPIO5_2,
Left,8,0,13,gfpaga_pad_IO_F2A[1293],F2A_GPIO5_1,
Left,8,0,12,gfpaga_pad_IO_F2A[1292],F2A_GPIO5_0,
Left,8,0,11,gfpaga_pad_IO_A2F[1291],A2F_GPIO5_3,
Left,8,0,10,gfpaga_pad_IO_A2F[1290],A2F_GPIO5_2,
Left,8,0,9,gfpaga_pad_IO_A2F[1289],A2F_GPIO5_1,
Left,8,0,8,gfpaga_pad_IO_A2F[1288],A2F_GPIO5_0,
Left,8,0,7,gfpaga_pad_IO_F2A[1287],F2A_IXS15_SDI_HS,
Left,8,0,6,gfpaga_pad_IO_F2A[1286],F2A_IXS14_SDI_HS,
Left,8,0,5,gfpaga_pad_IO_F2A[1285],F2A_IXS13_SDI_HS,
Left,8,0,4,gfpaga_pad_IO_F2A[1284],F2A_IXS12_SDI_HS,
Left,8,0,3,gfpaga_pad_IO_F2A[1283],F2A_IXS11_SDI_HS,
Left,8,0,2,gfpaga_pad_IO_F2A[1282],F2A_IXS10_SDI_HS,
Left,8,0,1,gfpaga_pad_IO_F2A[1281],F2A_IXS9_SDI_HS,
Left,8,0,0,gfpaga_pad_IO_F2A[1280],F2A_IXS8_SDI_HS,
Left,7,0,15,gfpaga_pad_IO_F2A[1279],F2A_GPIO4_31,
Left,7,0,14,gfpaga_pad_IO_F2A[1278],F2A_GPIO4_30,
Left,7,0,13,gfpaga_pad_IO_F2A[1277],F2A_GPIO4_29,
Left,7,0,12,gfpaga_pad_IO_F2A[1276],F2A_GPIO4_28,
Left,7,0,11,gfpaga_pad_IO_A2F[1275],A2F_GPIO4_31,
Left,7,0,10,gfpaga_pad_IO_A2F[1274],A2F_GPIO4_30,
Left,7,0,9,gfpaga_pad_IO_A2F[1273],A2F_GPIO4_29,
Left,7,0,8,gfpaga_pad_IO_A2F[1272],A2F_GPIO4_28,
Left,7,0,7,gfpaga_pad_IO_F2A[1271],RFU,
Left,7,0,6,gfpaga_pad_IO_F2A[1270],RFU,
Left,7,0,4,gfpaga_pad_IO_F2A[1268],F2A_AC_IXS_RX_FCLK_HS,
Left,7,0,2,gfpaga_pad_IO_A2F[1266],A2F_AC_FCLK_DIV_F_OUT_HS,
Left,7,0,1,gfpaga_pad_IO_A2F[1265],A2F_AC_FCLK_FMT_F_OUT_HS,
Left,7,0,0,gfpaga_pad_IO_A2F[1264],A2F_IXS_TX_FCLK_HS,
Left,6,0,15,gfpaga_pad_IO_F2A[1263],F2A_GPIO4_27,
Left,6,0,14,gfpaga_pad_IO_F2A[1262],F2A_GPIO4_26,
Left,6,0,13,gfpaga_pad_IO_F2A[1261],F2A_GPIO4_25,
Left,6,0,12,gfpaga_pad_IO_F2A[1260],F2A_GPIO4_24,
Left,6,0,11,gfpaga_pad_IO_A2F[1259],A2F_GPIO4_27,
Left,6,0,10,gfpaga_pad_IO_A2F[1258],A2F_GPIO4_26,
Left,6,0,9,gfpaga_pad_IO_A2F[1257],A2F_GPIO4_25,
Left,6,0,8,gfpaga_pad_IO_A2F[1256],A2F_GPIO4_24,
Left,6,0,7,gfpaga_pad_IO_F2A[1255],RFU,
Left,6,0,6,gfpaga_pad_IO_F2A[1254],RFU,
Left,6,0,4,gfpaga_pad_IO_F2A[1252],F2A_AC_IXS_TX_FCLK_HS,
Left,6,0,3,gfpaga_pad_IO_A2F[1251],A2F_AC_FCLK_0B0_IN_HS,

```

(continues on next page)

(continued from previous page)

```

Left,6,0,1,gfpga_pad_IO_A2F[1249],A2F_AC_FCLK_FMT_F_IN_HS,
Left,6,0,0,gfpga_pad_IO_A2F[1248],A2F_IXS_RX_FCLK_HS,
Left,5,0,15,gfpga_pad_IO_F2A[1247],F2A_GPIO4_23,
Left,5,0,14,gfpga_pad_IO_F2A[1246],F2A_GPIO4_22,
Left,5,0,13,gfpga_pad_IO_F2A[1245],F2A_GPIO4_21,
Left,5,0,12,gfpga_pad_IO_F2A[1244],F2A_GPIO4_20,
Left,5,0,11,gfpga_pad_IO_A2F[1243],A2F_GPIO4_23,
Left,5,0,10,gfpga_pad_IO_A2F[1242],A2F_GPIO4_22,
Left,5,0,9,gfpga_pad_IO_A2F[1241],A2F_GPIO4_21,
Left,5,0,8,gfpga_pad_IO_A2F[1240],A2F_GPIO4_20,
Left,5,0,3,gfpga_pad_IO_A2F[1235],A2F_AC_FCLK_0B1_IN_HS,
Left,5,0,0,gfpga_pad_IO_F2A[1232],RFU,
Left,4,0,15,gfpga_pad_IO_F2A[1231],F2A_GPIO4_19,
Left,4,0,14,gfpga_pad_IO_F2A[1230],F2A_GPIO4_18,
Left,4,0,13,gfpga_pad_IO_F2A[1229],F2A_GPIO4_17,
Left,4,0,12,gfpga_pad_IO_F2A[1228],F2A_GPIO4_16,
Left,4,0,11,gfpga_pad_IO_A2F[1227],A2F_GPIO4_19,
Left,4,0,10,gfpga_pad_IO_A2F[1226],A2F_GPIO4_18,
Left,4,0,9,gfpga_pad_IO_A2F[1225],A2F_GPIO4_17,
Left,4,0,8,gfpga_pad_IO_A2F[1224],A2F_GPIO4_16,
Left,4,0,5,gfpga_pad_IO_A2F[1221],A2F_GPIO5_WR_ACK,
Left,4,0,4,gfpga_pad_IO_A2F[1220],A2F_GPIO5_RD_ACK,
Left,4,0,3,gfpga_pad_IO_F2A[1219],F2A_AC_IXS_RX_BCLK,
Left,4,0,2,gfpga_pad_IO_F2A[1218],F2A_ACM_AES3_RX_BCLK,
Left,4,0,1,gfpga_pad_IO_F2A[1217],F2A_ACM_AES2_RX_BCLK,
Left,4,0,0,gfpga_pad_IO_F2A[1216],F2A_AC_AES_TX_MCLK,
Left,3,0,15,gfpga_pad_IO_F2A[1215],F2A_GPIO4_15,
Left,3,0,14,gfpga_pad_IO_F2A[1214],F2A_GPIO4_14,
Left,3,0,13,gfpga_pad_IO_F2A[1213],F2A_GPIO4_13,
Left,3,0,12,gfpga_pad_IO_F2A[1212],F2A_GPIO4_12,
Left,3,0,11,gfpga_pad_IO_A2F[1211],A2F_GPIO4_15,
Left,3,0,10,gfpga_pad_IO_A2F[1210],A2F_GPIO4_14,
Left,3,0,9,gfpga_pad_IO_A2F[1209],A2F_GPIO4_13,
Left,3,0,8,gfpga_pad_IO_A2F[1208],A2F_GPIO4_12,
Left,3,0,1,gfpga_pad_IO_F2A[1201],F2A_ACM_AES2_RX_FCLK,
Left,3,0,0,gfpga_pad_IO_F2A[1200],F2A_AC_ADLL_IN,
Left,2,0,15,gfpga_pad_IO_F2A[1199],F2A_GPIO4_11,
Left,2,0,14,gfpga_pad_IO_F2A[1198],F2A_GPIO4_10,
Left,2,0,13,gfpga_pad_IO_F2A[1197],F2A_GPIO4_9,
Left,2,0,12,gfpga_pad_IO_F2A[1196],F2A_GPIO4_8,
Left,2,0,11,gfpga_pad_IO_A2F[1195],A2F_GPIO4_11,
Left,2,0,10,gfpga_pad_IO_A2F[1194],A2F_GPIO4_10,
Left,2,0,9,gfpga_pad_IO_A2F[1193],A2F_GPIO4_9,
Left,2,0,8,gfpga_pad_IO_A2F[1192],A2F_GPIO4_8,
Left,2,0,1,gfpga_pad_IO_F2A[1185],F2A_ACM_AES3_RX_FCLK,
Left,2,0,0,gfpga_pad_IO_F2A[1184],F2A_AC_IXS_TX_BCLK,
Left,1,0,15,gfpga_pad_IO_F2A[1183],F2A_GPIO4_7,
Left,1,0,14,gfpga_pad_IO_F2A[1182],F2A_GPIO4_6,
Left,1,0,13,gfpga_pad_IO_F2A[1181],F2A_GPIO4_5,
Left,1,0,12,gfpga_pad_IO_F2A[1180],F2A_GPIO4_4,
Left,1,0,11,gfpga_pad_IO_A2F[1179],A2F_GPIO4_7,
Left,1,0,10,gfpga_pad_IO_A2F[1178],A2F_GPIO4_6,

```

(continues on next page)



(continued from previous page)

```

Left,1,0,9,gfpga_pad_IO_A2F[1177],A2F_GPIO4_5,
Left,1,0,8,gfpga_pad_IO_A2F[1176],A2F_GPIO4_4,
Left,0,0,15,gfpga_pad_IO_F2A[1167],F2A_GPIO4_3,
Left,0,0,14,gfpga_pad_IO_F2A[1166],F2A_GPIO4_2,
Left,0,0,13,gfpga_pad_IO_F2A[1165],F2A_GPIO4_1,
Left,0,0,12,gfpga_pad_IO_F2A[1164],F2A_GPIO4_0,
Left,0,0,11,gfpga_pad_IO_A2F[1163],A2F_GPIO4_3,
Left,0,0,10,gfpga_pad_IO_A2F[1162],A2F_GPIO4_2,
Left,0,0,9,gfpga_pad_IO_A2F[1161],A2F_GPIO4_1,
Left,0,0,8,gfpga_pad_IO_A2F[1160],A2F_GPIO4_0,

```

**Syntax:****To set constraint for the IO**

```
set_io <port_name> <Package IO>
```

**To set constraint for the clock port**

```
set_clk <clk_name> <port_name>
```

clock names supported: clk[0], clk[1], clk[2], clk[3]

For this csv file, the counter\_16bit one can add these IO placements:

```

set_io enable user_IN_T[0]
set_io reset user_IN_T[1]
set_clk clk[0] clk
set_io count[0] user_OUT_T[8]
set_io count[1] user_OUT_T[9]
set_io count[2] user_OUT_T[10]
set_io count[3] user_OUT_T[11]
set_io count[4] user_OUT_T[12]
set_io count[5] user_OUT_T[13]
set_io count[6] user_OUT_T[14]
set_io count[7] user_OUT_T[15]
set_io count[8] user_OUT_T[16]
set_io count[9] user_OUT_T[17]
set_io count[10] user_OUT_T[18]
set_io count[11] user_OUT_T[19]
set_io count[12] user_OUT_T[20]
set_io count[13] user_OUT_T[21]
set_io count[14] user_OUT_T[22]
set_io count[15] user_OUT_T[23]

```

**Note:** > The pcf file support is only supported for the QLF\_K4N8

## 2.4 BRAM and DSP

To infer BRAM:

In SymbiFlow, the synthesis of Verilog files is performed with Yosys which infers the BRAM for the device qlf\_k6n10.

Syntax:

```
ql_symbiflow -synth -src <source complete path> -d <device> -t <top_
module name> -v <verilog files>
```

Output files for synthesis are present in 'build' folder:

<TOP>.eblif : netlist file for the design <TOP>\_synth.log : synthesis log information, refer this file for any issues during synthesis

Resource utilization in the top\_synth.log of the BRAM design:

```
io          : 42
  inpad     : 26
  outpad    : 16
clb         : 2
  fle       : 2
  lut5inter : 2
  ble5      : 2
  flut5     : 2
  lut5      : 2
  lut       : 2
memory      : 1
  mem_16K_dp : 1
```

All the output log files will be dumped in {source path}/build folder

```
//Verilog code for the BRAM configuration of 16x1024
module BRAM_16x1024_TOP( clk, rce, ra, rq, wce, wa, wd);

parameter AWIDTH = 10;
parameter DWIDTH = 16;
input  clk;
input  rce;
input  [AWIDTH-1:0] ra;
input  wce;
input  [AWIDTH-1:0] wa;
input  [DWIDTH-1:0] wd;
output [DWIDTH-1:0] rq;

    BRAM #(.AWIDTH(AWIDTH), .DWIDTH(DWIDTH)) BRAM_16x1024 (.
clk(clk), .rce(rce), .ra(ra), .rq(rq), .wce(wce), .wa(wa), .wd(wd));

endmodule

module BRAM #(parameter AWIDTH = 9,
parameter DWIDTH = 32)( clk, rce, ra, rq, wce, wa, wd);
```

(continues on next page)

(continued from previous page)

```

input  clk;
input  rce;
input  wce;
input  [AWIDTH-1:0] ra;
input  [AWIDTH-1:0] wa;
input  [DWIDTH-1:0] wd;
output reg [DWIDTH-1:0] rq;
reg    [DWIDTH-1:0] memory[0:AWIDTH-1];

always @(posedge clk) begin
    if (rce)
        rq <= memory[ra];
    if (wce)
        memory[wa] <= wd;
end

integer i;
initial
begin
    for(i = 0; i < AWIDTH-1; i = i + 1)
        memory[i] = 0;
end
endmodule

```

The various configuration of the BRAMS can be configured by changing the AWIDTH and DWIDTH parameters as shown below:

BRAM 4x4096 : AWIDTH=12, DWIDTH=4

BRAM 8x2048 : AWIDTH=11, DWIDTH=8

BRAM 32x512 : AWIDTH=10, DWIDTH=16

#### To infer DSP:

To infer the DSP block for multiplier, refer the mac\_16 verilog example.

Resource utilization in the top\_synth.log of the MAC\_16 design:

```

io           : 64
inpad        : 48
outpad       : 16
clb          : 1
fle          : 1
lut5inter    : 1
ble5         : 1
flut5        : 1
lut5         : 1
lut          : 1
QL_DSP       : 1
QL_DSP_slice : 1

```

All the output log files will be dumped in {source path}/build folder

```

//Verilog code for the multiplier+adder block
module mac_16(a, b, c, out);

```

(continues on next page)

(continued from previous page)

```
parameter DATA_WIDTH = 16;
input [DATA_WIDTH - 1 : 0] a, b, c;
output [DATA_WIDTH - 1 : 0] out;

assign out = a * b + c;

endmodule
```

---

**Note:** > The programming files are only supported for the **QLF\_K6N10**

---

## 2.5 Online References

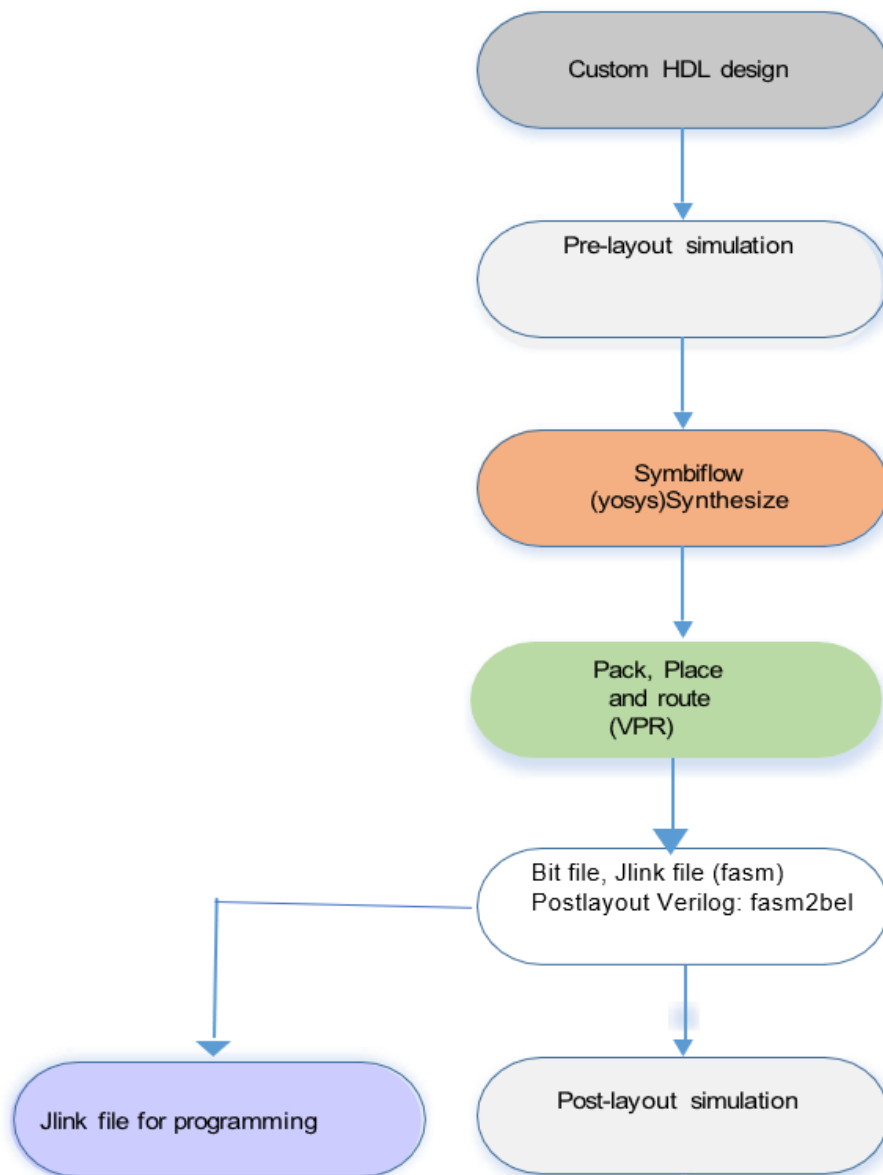
References: [SDC constraints information](#) [VPR flow and the files info](#)

## S3B DEVICE

This provides the details of running designs and in-built RAMs/FIFOs/Multipliers in S3B.

### 3.1 Symbiflow: Design flow (EOS-S3)

The below figure shows the complete design flow from the HDL to the programming file:



### 3.1.1 Supported Commands

Command option	Represented for	Options
<b>-synth</b>	Synthesis using yosys	-
<b>-compile</b>	Run pack, place, route and generate fasm file	-
<b>-src &lt;source path&gt;</b>	Source file folder	-
<b>-d &lt;device&gt;</b>	Device supported	ql-eos-s3
<b>-P &lt;package&gt;</b>	Packages	PD64 (BGA), WR42(WLCSP), PU64 (QFN)
<b>-p &lt;pcf file&gt;</b>	Fix Placement constraints of IO's	-
<b>-s &lt;sdc file&gt;</b>	Timing Constraint File (SDC)	Refer online documents section for SDC constraints supported
<b>-r &lt;router flag&gt;</b>	Timing: means no attention is paid to delay. Congestion: means nets on the critical path pay no attention to congestion.	timing, congestion
<b>-t &lt;top module&gt;</b>	Top module of the Verilog design	-
<b>-v &lt;Verilog list files&gt;</b>	Verilog source files	Only Verilog supported
<b>-dump &lt;output to be dumped&gt;</b>	Dump the output format file	Jlink, post_verilog, header

## 3.2 Run design flow on a simple counter design (EOS-S3)

Setup environment

To run any example, perform these steps once.

```
export INSTALL_DIR="specify the installpath"
#adding symbiflow toolchain binaries to PATH
export PATH="$INSTALL_DIR/quicklogic-arch-defs/bin:$INSTALL_DIR/
quicklogic-arch-defs/bin/python:$PATH"
source "$INSTALL_DIR/conda/etc/profile.d/conda.sh"
conda activate
```

Entering an HDL Design:

1. Write a Verilog code for the design using any text editor.
2. Verify the syntax.
3. Create the simulation stimuli using any text editor.

The code and testbench for the example design are present at: <Install\_Path>/quicklogic-arch-defs/tests/counter\_16bit/

### 3.2.1 Performing the Pre-Layout Simulation

To perform a pre-layout simulation:

#### Using Icarus Verilog:

To create the VCD output file that will be used to perform graphical analysis of the Design, the following lines are added in the TB:

```
initial begin
    $dumpfile("counter_16bit_tb.vcd");
    $dumpvars(0,counter_16bit_tb);
    $display("\t\ttime,\t\tclk,\t\treset,\t\ttenable,\t\ttcount");
    $monitor("%d,\t\tb,\t\tb,\t\tb,\t\tb", $time, clk,reset,enable,
    ↪count);
end
```

The “iverilog” and “vvp” commands are the most important commands available to users of Icarus Verilog. The “iverilog” command is the compiler, and the “vvp” command is the simulation runtime engine.

```
cd <INSTALL_PATH>/quicklogic-arch-defs/tests/counter_16bit
```

The “iverilog” command supports multi-file designs by two methods. The simplest is to list the files on the command line:

```
iverilog -o my_design counter_16bit.v counter_16bit_tb.v
vvp my_design
```

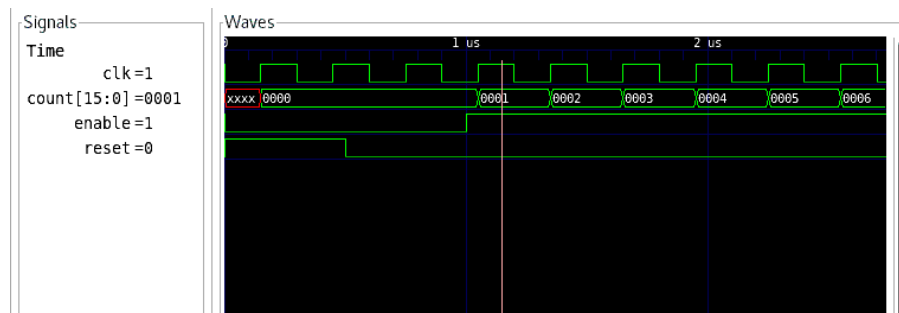
This command compiles the design, which is spread across two input files, and generates the compiled result into the “my\_design” file. Another technique is to use a commandfile, which lists the input files in a text file. For example, create a text file called “file\_list.txt” with the files listed one per line:

```
#Inside file_list.txt
counter_16bit.v
counter_16bit_tb.v
```

```
#Then compile and execute the design with command:
iverilog -o my_design -c file_list.txt
vvp my_design
```

VCD file is created, it can be viewed using GTKWave:

```
gtkwave testbench.vcd &
```





### 3.2.2 Performing Design Synthesis

To perform a design synthesis:

In SymbiFlow, the synthesis of Verilog files is performed with Yosys. Yosys parses Verilog files, applies basic optimizations, performs technological mapping to FPGA blocks, and generates JSON and EBLIF files for the place and route tool.

Syntax:

```
ql_symbiflow -synth -src <source complete path> -d <device> -t <top_
↳module name> -v <verilog files>
```

```
cd <INSTALL_PATH>/quicklogic-arch-defs/tests/counter_16bit
```

and run the below command:

```
ql_symbiflow -synth -d ql-eos-s3 -t top -v counter_16bit.v
```

Output files for synthesis are: <TOP>.eblif : netlist file for the design <TOP>\_synth.log : synthesis log information, refer this file for any issues during synthesis

Resource utilization in the top\_synth.log of the counter:

*Number of wires: 384 Number of wire bits: 384 Number of public wires: 382 Number of public wire bits: 382 Number of memories: 0 Number of memory bits: 0 Number of processes: 0 Number of cells: 67 BIDIR\_CELL 18 CLOCK\_CELL 1 C\_FRAG 4 GMUX\_IP 1 Q\_FRAG 16 T\_FRAG 27*

**Note:** > All the output log files will be dumped in {source path}/build folder > -src command is optional if run from the same directory where source files are present.

### 3.2.3 Running pack, Place and Route tools (EOS-S3)

The eblif file generated during the synthesis is used for pack, place and route along with device information, pcf and the source files.

Syntax:

```
ql_symbiflow -compile -src <source complete path> -d <device> -t <top_
↳module name> -v <verilog files> -p <pcf file> -P <Package file> -s <SDC_
↳file>
```

**The output files dumped will be:** <TOP>.net : Once packing is complete. <TOP>.place : Placer file from VPR <TOP>.route : Router file from VPR

One can refer to the pack.log, placer.log, router.log for more information related to each tool.

```
cd <INSTALL_PATH>/quicklogic-arch-defs/tests/counter_16bit
ql_symbiflow -compile -src $PWD -d ql-eos-s3 -t top -v counter_16bit.v -p_
↳chandalar.pcf -P PD64 -s counter_16bit.sdc
```

The above command will also run synthesis if it was not run before.

To Generate Various files during compile, use the below options Common command with just output file change:

```
ql_symbiflow -compile -src $PWD -d ql-eos-s3 -t top -v counter_16bit.v_
↪ -p chandalar.pcf -P PD64 -s counter_16bit.sdc -dump jlink/post_
↪ verilog/header
```

To Generate the Post-Layout Verilog file

This is the Verilog file used for the functional simulation to verify the Place and Route output.

Syntax:

```
ql_symbiflow -compile -src <source complete path> -d <device> -t <top_
↪ module name> -v <verilog files> -p <pcf file> -P <Package file> -s
↪ <SDC file> -dump post_verilog
```

The output files dumped will be:

top\_bit.v : Post layout Verilog file (for verifying the configuration bits) top\_post\_synthesis.v  
: Post layout Verilog file (for timing simulation) top\_post\_synthesis.sdf : SDF file (for timing  
simulation)

```
ql_symbiflow -compile -src $PWD -d ql-eos-s3 -t top -v counter_16bit.v_
↪ -p chandalar.pcf -P PD64 -s counter_16bit.sdc -dump post_verilog
```

The Timing analysis refer the files *report\_timing.hold.rpt*, *report\_timing.setup.rpt* and *top.log* inside the build folder

For the counter design below is the timing report from the top.log file:

*Hold Worst Negative Slack (hWNS): 0 ns Hold Total Negative Slack (hTNS): 0 ns Setup Worst  
Negative Slack (sWNS): -35.2295 ns Setup Total Negative Slack (sTNS): -1062.29 ns Final crit-  
ical path: 35.2295 ns, Fmax: 28.3853 MHz*

### 3.2.4 Performing the Post-Layout Simulation (verifying the configuration bits)

Post layout Simulation using the configuration bits, uses the device configuration bit file *top\_bit.v*

The testbench for the counter design is present at: *<Install\_Path>/quicklogic-arch-  
defs/tests/counter\_16bit/*

The post-layout design netlist is present at: *<Install\_Path>/quicklogic-arch-  
defs/counter\_16bit/top\_bit.v*

The primitive file library file is present at: *<Install\_Path>/conda/share/yosys/quicklogic/cells\_sim.v*

---

**Note:** cells\_sim.v : This file has the definition for predefined macros

---

### 3.2.5 Performing the Post-Layout Timing Simulation

Post layout Timing simulation uses the SDF(Standard Delay Format) file.

The testbench for the counter design is present at: `<Install_Path>/quicklogic-arch-defs/tests/counter_16bit/`

The post-layout design netlist is present at: `<Install_Path>/quicklogic-arch-defs/counter_16bit/build/top_post_synthesis.v`

The SDF file is present at: `<Install_Path>/quicklogic-arch-defs/counter_16bit/build/top_post_synthesis.sdf`

The primitive file library file is present at: `<Install_Path>/quicklogic-arch-defs/share/techmaps/quicklogic/techmap/cells_sim.v`

The ram primitive file is present at: `<Install_Path>/quicklogic-arch-defs/share/arch/ql-eos-s3_wlcsp/cells/ram_sim.v`

**To perform a post-layout simulation:** - Perform a post-layout simulation of the Verilog code use iverilog. - View the simulation results in the Waveform/ Data Analyzer and verify.

---

**Note:** cells\_sim.v : This file has the definition for technology mapped macros ram\_sim.v : Has the ram definition

---

### 3.2.6 Generate the Jlink and openOCD file

**JLINK file contains a script that can flash the board with the generated FPGA configuration via the JLink Connector**

Syntax:

```
ql_symbiflow -compile -src <source complete path> -d <device> -t <top_
module name> -v <verilog files> -p <pcf file> -P <Package file> -s <SDC_
file> -dump jlink
```

**The output files dumped will be:** <TOP>.jlink ->jlink file.

```
ql_symbiflow -compile -src $PWD -d ql-eos-s3 -t top -v counter_16bit.v -p_
chandalar.pcf -P PD64 -s counter_16bit.sdc -dump jlink
```

For details on how to configure the FPGA using the top.jlink file, refer to Download Binaries using Jlink SWD section in the [QuickFeather\\_UserGuide](#) pdf.

openOCD is an on-chip debugger file Syntax:

```
ql_symbiflow -compile -src <source complete path> -d <device> -t <top_
module name> -v <verilog files> -p <pcf file> -P <Package file> -s
<SDC file> -dump openocd
```

**The output files dumped will be:** <TOP>.openocd ->openOCD file.

```
ql_symbiflow -compile -src $PWD -d ql-eos-s3 -t top -v counter_16bit.v -p_
chandalar.pcf -P PD64 -s counter_16bit.sdc -dump openocd
```

### 3.2.7 Generate the ASCII header file format

Ascii header file can be generated from the jlink or the .bit file.

Syntax:

```
ql_symbiflow -compile -src <source complete path> -d <device> -  
↳t <top module name> -v <verilog files> -p <pcf file> -P  
↳<Package file> -s <SDC file> -dump header
```

**The output files dumped will be :** <TOP>\_bit.h - *file generated from the bit file input*

```
ql_symbiflow -compile -src $PWD -d ql-eos-s3 -t top -v counter_  
↳16bit.v -p chandalar.pcf -P PD64 -s counter_16bit.sdc -dump_  
↳header
```

**The generated header file can be used in M4 application program to load FPGA**

The output files can be dumped for all as:

```
ql_symbiflow -compile -src $PWD -d ql-eos-s3 -t top -v counter_  
↳16bit.v -p chandalar.pcf -P PD64 -s counter_16bit.sdc -dump_  
↳header jlink post_verilog
```

### 3.2.8 Generate the Binary File format

FPGA Binary file can be generated, which will contain the following components:

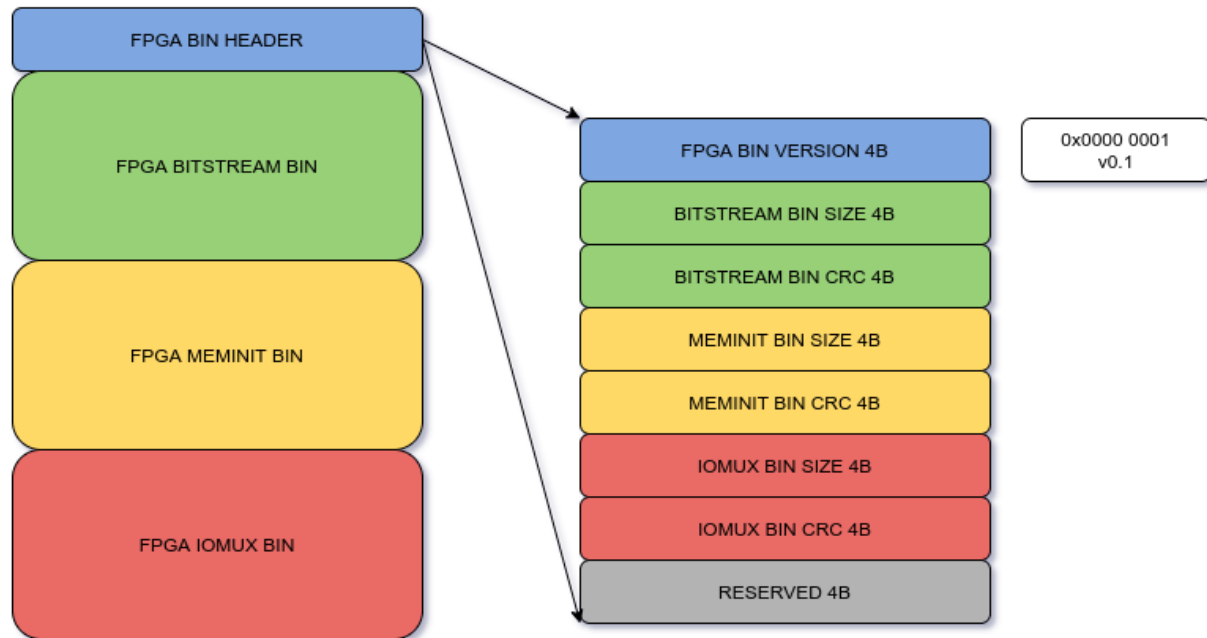
1. FPGA Bitstream
2. FPGA MemInit (RAM Block Initialization)
3. FPGA IOMux Configuration

Then, FPGA bin [\${TOP}.bin] == bitstream bin + meminit bin + iomux bin

We will need a header in the FPGA bin which indicates the component information (sizes).

The diagrams below show a complete picture of the FPGA bin structure.

## FPGA bin header

**FPGA BINARY STRUCTURE****HEADER**

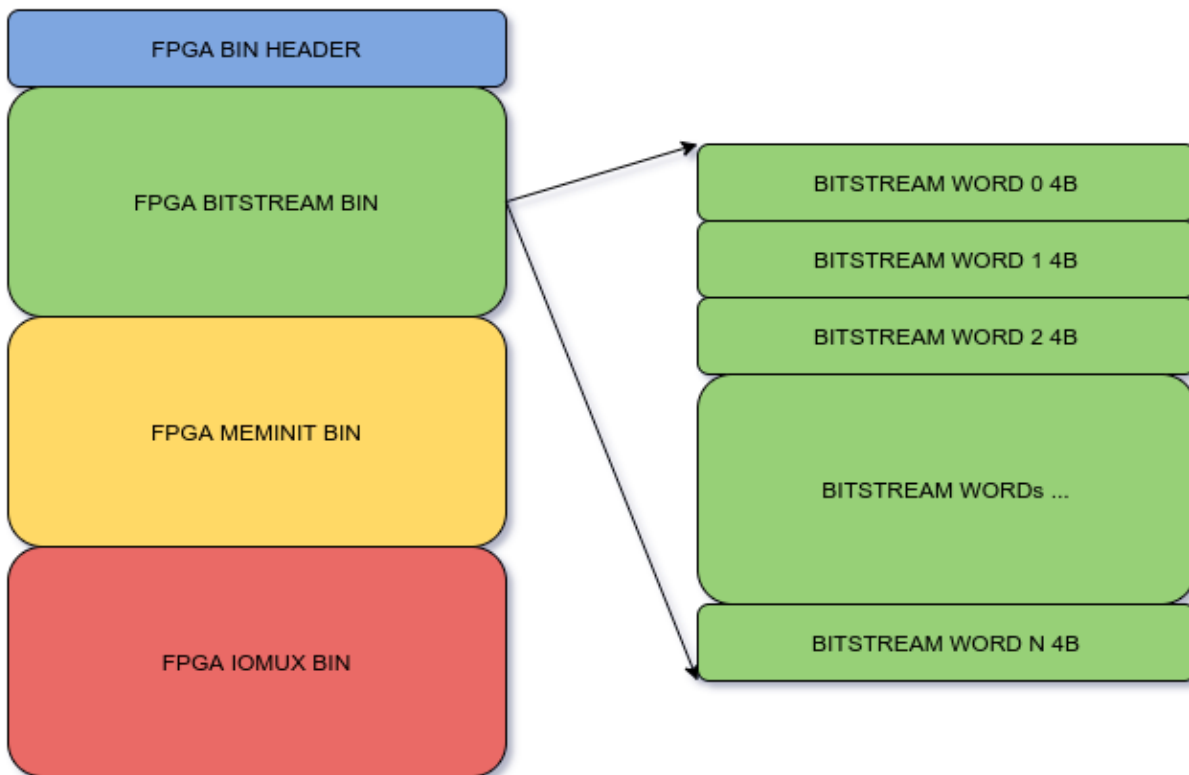
The header contains the fields:

- FPGA BIN VERSION - which will help handle future changes in bin structure if needed. Currently at v0.1.
- BITSTREAM BIN SIZE, CRC - size in bytes, and crc of the bitstream binary content
- MEMINIT BIN SIZE, CRC - size in bytes, and crc of the meminit binary content
- IOMUX BIN SIZE, CRC - size in bytes, and crc of the iomux binary content
- RESERVED WORD - for future usage

## FPGA bitstream bin

## FPGA BINARY STRUCTURE

### BITSTREAM

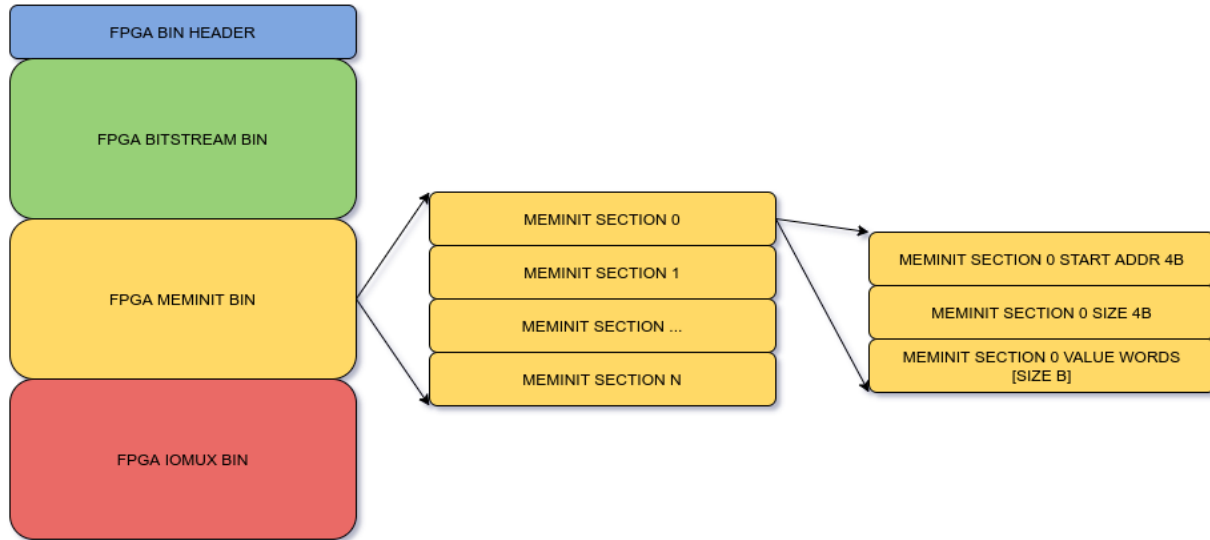


The bitstream bin will have 4B words, exactly as is currently generated in `${TOP}.bit`

## FPGA meminit bin

## FPGA BINARY STRUCTURE

## MEMINIT



The meminit bin will have the same structure as is currently generated in the header method.

For each RAM block, we will have:

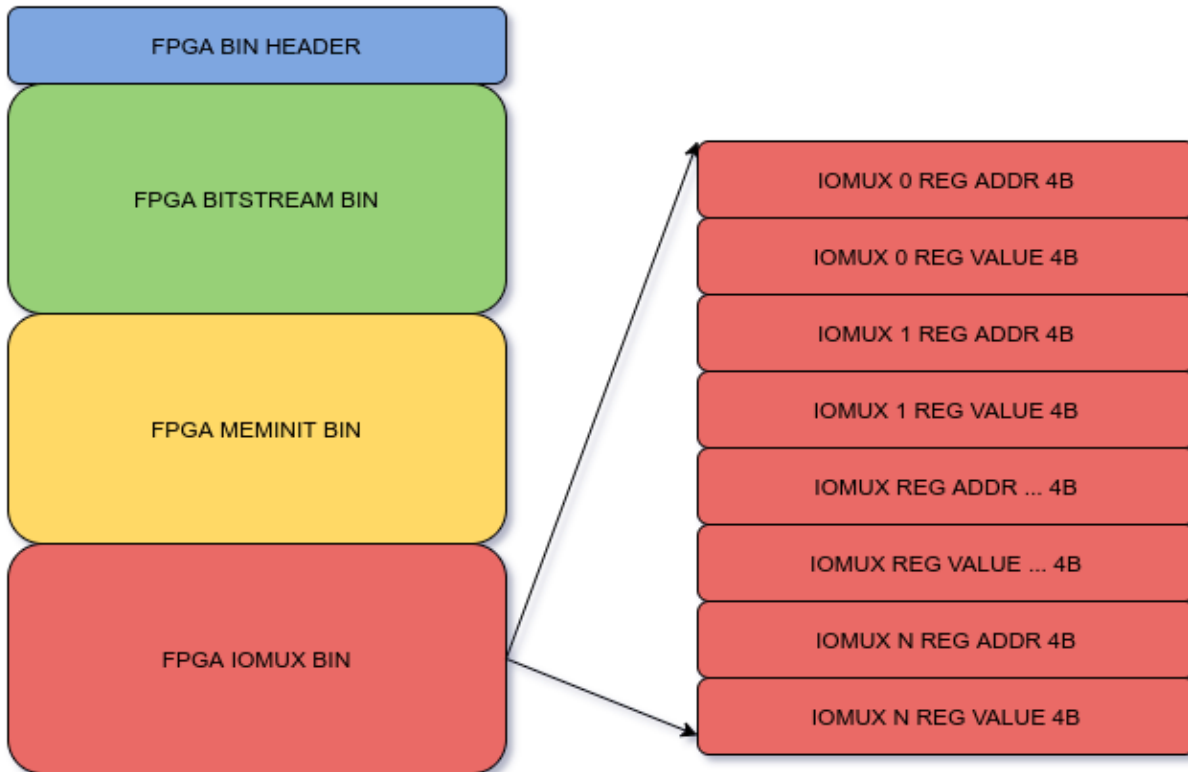
- RAM block start address 4B
- RAM block size 4B
- size B of initialization values

This set will be repeated for as many RAM blocks initialized in the design.

## FPGA iomux bin

# FPGA BINARY STRUCTURE

## IOMUX



The iomux bin will have a set of pad configurations, each pad configuration is:

- 4B Reg Address
- 4B Reg Value

The number of pad configurations would be equal to the number of pads used by the FPGA design.

Syntax:

```
ql_symbiflow -compile -src <source complete path> -d <device> -t <top module name> -v
  ↳<verilog files> -p <pcf file> -P <Package file> -s <SDC file> -dump binary
```

The output files dumped will be :

<TOP>.bin - file generated from the bit file input

```
ql_symbiflow -compile -src $PWD -d ql-eos-s3 -t top -v counter_16bit.v -p chandalar.pcf -
  ↳P PD64 -s counter_16bit.sdc -dump binary
```



The generated binary file can then be flashed into the board with the programmer, and the bootloader can automatically load the FPGA design.

### 3.2.9 PCF Sample

The PCF file is for fix placing the IO to a particular IO location on the device. For S3B device we have 3 packages PD64, PU64 and WR42.

For package PD64, the counter\_16bit has the below IO placements: Syntax: set\_io <port\_name> <Package IO>

```
set_io clk A3
set_io enable C1
set_io reset A1
set_io count(0) A2
set_io count(1) B2
set_io count(2) C3
set_io count(3) B3
set_io count(4) B1
set_io count(5) C4
set_io count(6) B4
set_io count(7) A4
set_io count(8) C5
set_io count(9) B5
set_io count(10) D6
set_io count(11) A5
set_io count(12) C6
set_io count(13) E7
set_io count(14) D7
set_io count(15) E8
```

PCF reference file for various the below packages

The highlighted pins are the clock ports and can also be used as BIDIR IO. Either IO Location or Alias name can be used.

PD64		
IO Locatio	Alias	IO Type
B1	IO_0	BIDIR
C1	IO_1	BIDIR
A1	IO_2	BIDIR
A2	IO_3	BIDIR
B2	IO_4	BIDIR
C3	IO_5	BIDIR
B3	IO_6	BIDIR
A3	IO_7	BIDIR/CLOCK
C4	IO_8	BIDIR/CLOCK
B4	IO_9	BIDIR
A4	IO_10	BIDIR
C5	IO_11	BIDIR
B5	IO_12	BIDIR
D6	IO_13	BIDIR
A5	IO_14	BIDIR
C6	IO_15	BIDIR
E7	IO_16	BIDIR
D7	IO_17	BIDIR
E8	IO_18	BIDIR
H8	IO_19	BIDIR
G8	IO_20	BIDIR
H7	IO_21	BIDIR
G7	IO_22	BIDIR/CLOCK
H6	IO_23	BIDIR/CLOCK
G6	IO_24	BIDIR/CLOCK
F7	IO_25	BIDIR
F6	IO_26	BIDIR
H5	IO_27	BIDIR
G5	IO_28	BIDIR
F5	IO_29	BIDIR
F4	IO_30	BIDIR
G4	IO_31	BIDIR
H4	IO_32	SDIOMUX
E3	IO_33	SDIOMUX
F3	IO_34	SDIOMUX
F2	IO_35	SDIOMUX
H3	IO_36	SDIOMUX
G2	IO_37	SDIOMUX
E2	IO_38	SDIOMUX
H2	IO_39	SDIOMUX
D2	IO_40	SDIOMUX
F1	IO_41	SDIOMUX
H1	IO_42	SDIOMUX
D1	IO_43	SDIOMUX
E1	IO_44	SDIOMUX
G1	IO_45	SDIOMUX

PU64		
IO Locatio	Alias	IO type
4	IO_0	BIDIR
5	IO_1	BIDIR
6	IO_2	BIDIR
2	IO_3	BIDIR
3	IO_4	BIDIR
64	IO_5	BIDIR
62	IO_6	BIDIR
63	IO_7	BIDIR/CLOCK
61	IO_8	BIDIR/CLOCK
60	IO_9	BIDIR
59	IO_10	BIDIR
57	IO_11	BIDIR
56	IO_12	BIDIR
55	IO_13	BIDIR
54	IO_14	BIDIR
53	IO_15	BIDIR
40	IO_16	BIDIR
42	IO_17	BIDIR
38	IO_18	BIDIR
36	IO_19	BIDIR
37	IO_20	BIDIR
39	IO_21	BIDIR
34	IO_22	BIDIR/CLOCK
33	IO_23	BIDIR/CLOCK
32	IO_24	BIDIR/CLOCK
31	IO_25	BIDIR
30	IO_26	BIDIR
28	IO_27	BIDIR
27	IO_28	BIDIR
26	IO_29	BIDIR
25	IO_30	BIDIR
23	IO_31	BIDIR
22	IO_32	SDIOMUX
21	IO_33	SDIOMUX
20	IO_34	SDIOMUX
18	IO_35	SDIOMUX
17	IO_36	SDIOMUX
15	IO_37	SDIOMUX
16	IO_38	SDIOMUX
11	IO_39	SDIOMUX
13	IO_40	SDIOMUX
14	IO_41	SDIOMUX
10	IO_42	SDIOMUX
7	IO_43	SDIOMUX
8	IO_44	SDIOMUX
9	IO_45	SDIOMUX

WR42		
IO Locatio	Alias	IO Type
A7	IO_0	BIDIR
B7	IO_1	BIDIR
C7	IO_3	BIDIR
A6	IO_6	BIDIR
B6	IO_8	BIDIR/CLOCK
A5	IO_9	BIDIR
B5	IO_10	BIDIR
A4	IO_14	BIDIR
B4	IO_15	BIDIR
E1	IO_16	BIDIR
D1	IO_17	BIDIR
C1	IO_19	BIDIR
F2	IO_20	BIDIR
E2	IO_23	BIDIR/CLOCK
D2	IO_24	BIDIR/CLOCK
D3	IO_25	BIDIR
F3	IO_28	BIDIR
E3	IO_29	BIDIR
F4	IO_30	BIDIR
E4	IO_31	BIDIR
D5	IO_34	SDIOMUX
F5	IO_36	SDIOMUX
E6	IO_38	SDIOMUX
F6	IO_39	SDIOMUX
D7	IO_43	SDIOMUX
E7	IO_44	SDIOMUX
F7	IO_45	SDIOMUX

### 3.3 Hardware Limitations and Online References

Limitations:

- RAM/FIFO: Only symmetrical combinations of RAM are supported

References: [SDC constraints information](#) [VPR flow and the files info](#)

### 3.4 S3B Device

Hardmacro Resources

- RAMs - 8 blocks of 8K bits
- FIFO - 8 in-built FIFO controllers - Can be configured into FIFOs using above RAM blocks
- Multipliers - 2 (32X32) in-built multipliers

#### 3.4.1 RAM Features

- 8Kbits/RAM
- X8, x16 and x32 data bus width
- Independent programmability of read and write data bus widths
- Asynchronous clocks
- Supports 2 RAM block vertical or horizontal concatenation((i.e. 16Kbits RAM combining 2 8Kbits RAM Blocks)
- Supports clock disabling during idle operation

#### 3.4.2 FIFO Features

- X8, x16 and x32 data bus width
- Configurable Synchronous/Asynchronous operation
- Supports 2 RAM block vertical or horizontal concatenation((i.e. 16Kbits RAM combining 2 - 8Kbits RAM Blocks)
- Independent programmability of data bus width on PUSH and POP side
- Switchable clock domain between PUSH and POP side during asynchronous operation
- Supports clock disabling during idle operation
- Asynchronous Reset (aside from the synchronous FLUSH) going to the pointers

### 3.4.3 Multiplier Features

- 2 (32-bit) Multiplier
- Can be configured as 4 16-bit multipliers
- Signed multiplier
- Supports Latched input

### 3.4.4 Macro Usage and examples

#### RAM Usage

The RAM macros (RAM\_8K\_BLK.v & RAM\_16K\_BLK.v) are part of pp3\_cells\_sim.v at: *<Install\_Path>/conda/share/yosys/quicklogic*

The examples of RAMs are present at: *<Install\_Path>/quicklogic-arch-defs/tests/RAM\_Examples*

The Design example using S3 SRAM block are present at: *<Install\_Path>/quicklogic-arch-defs/tests/ram\_test*

1. RAM\_8K\_BLK RAM macro is 8K bits block which can be configured as:
  - 1 independent 8Kbits RAM block
  - RAM can be initialized through INIT or INIT\_FILE
2. RAM\_16K\_BLK RAM macro is a 16K bits block which can be configured as:
  - A horizontal concatenated block (16Kbits)
  - A vertical concatenated block (16Kbits)
  - RAM can be initialized through INIT or INIT\_FILE

Example 1: 1 8Kbits RAM

```
module r512x16_512x16 (WA,RA,WD,WClk,RClk,WClk_En,RClk_En,WEN,RD);
    input [8:0] WA;
    input [8:0] RA;
    input WClk,RClk;
    input WClk_En,RClk_En;
    input [1:0] WEN;
    input [15:0] WD;
    output [15:0] RD;

    parameter [16383:0] INIT = 16384'b0;
    parameter INIT_FILE="init_512x16.hex";

    parameter addr_int = 9 ;
    parameter data_depth_int = 512;
    parameter data_width_int = 16;
    parameter wr_enable_int = 2;
    parameter reg_rd_int = 0;

    RAM_8K_BLK #(.addr_int(addr_int),.data_depth_int(data_depth_int),.data_width_
    ↪int(data_width_int),.wr_enable_int(wr_enable_int),.reg_rd_int(reg_rd_int),
    .INIT(INIT),.INIT_FILE(INIT_FILE)
    )
```

(continues on next page)

(continued from previous page)

```

    RAM_INST (      .WA(WA), .RA(RA), .WD(WD), .WClk(WClk), .RClk(RClk), .WClk_En(WClk_
↪En), .RClk_En(RClk_En), .WEN(WEN), .RD(RD));
endmodule

```

## Example 2: Horizontal Concatenation

- 512x32 RAM (16 Kbits)
- RAM can be initialized through INIT or INIT\_FILE

```

module r512x32_512x32 (WA,RA,WD,WClk,RClk,WClk_En,RClk_En,WEN,RD);
    input [8:0] WA;
    input [8:0] RA;
    input WClk,RClk;
    input WClk_En,RClk_En;
    input [3:0] WEN;
    input [31:0] WD;
    output [31:0] RD;

    parameter [16383:0] INIT = 16384'b0;
    parameter INIT_FILE="init_512x32.hex";

    parameter addr_int = 9 ;
    parameter data_depth_int = 512;
    parameter data_width_int = 32;
    parameter wr_enable_int = 4;
    parameter reg_rd_int = 0;

    RAM_16K_BLK #(.addr_int(addr_int),.data_depth_int(data_depth_int),.data_width_
↪int(data_width_int),.wr_enable_int(wr_enable_int),.reg_rd_int(reg_rd_int),
                    .INIT(INIT),.INIT_FILE(INIT_FILE)
                )
    RAM_INST (      .WA(WA), .RA(RA), .WD(WD), .WClk(WClk), .RClk(RClk), .WClk_En(WClk_
↪En), .RClk_En(RClk_En), .WEN(WEN), .RD(RD));
endmodule

```

## Example 3: Vertical Concatenation

- 2048 x 8 RAM (16 Kbits)
- RAM can be initialized through INIT or INIT\_FILE

```

module r2048x8_2048x8 (WA,RA,WD,WClk,RClk,WClk_En,RClk_En,WEN,RD);
    input [10:0] WA;
    input [10:0] RA;
    input WClk,RClk;
    input WClk_En,RClk_En;
    input WEN;
    input [7:0] WD;
    output [7:0] RD;

    parameter [16383:0] INIT = 16384'b0;
    parameter INIT_FILE="init_2048x8.hex";

```

(continues on next page)

(continued from previous page)

```

parameter addr_int = 11 ;
parameter data_depth_int = 2048;
parameter data_width_int = 8;
parameter wr_enable_int = 1;
parameter reg_rd_int = 0;

RAM_16K_BLK #(.addr_int(addr_int),.data_depth_int(data_depth_int),.data_width_
↪int(data_width_int),.wr_enable_int(wr_enable_int),.reg_rd_int(reg_rd_int),
    .INIT(INIT),.INIT_FILE(INIT_FILE)
    )
RAM_INST (    .WA(WA), .RA(RA), .WD(WD), .WClk(WClk), .RClk(RClk), .WClk_En(WClk_
↪En), .RClk_En(RClk_En), .WEN(WEN), .RD(RD));
endmodule

```

## FIFO Usage

The FIFO macros (FIFO\_8K\_BLK.v & FIFO\_16K\_BLK.v) are part of pp3\_cells\_sim.v at: *<Install\_Path>/conda/share/yosys/quicklogic*

The examples of FIFOs are present at: *<Install\_Path>/quicklogic-arch-defs/tests/FIFO\_Examples*

The Design example using S3 FIFO block are present at: *<Install\_Path>/quicklogic-arch-defs/tests/fifo\_test*

Example 1: 1 8Kbits FIFO

- FIFO 512 x 16
- Asynchronous FIFO

```

module af512x16_512x16 (DIN,Fifo_Push_Flush,Fifo_Pop_Flush,PUSH,POP,Push_Clk,Pop_Clk,
↪Push_Clk_En,Pop_Clk_En,Fifo_Dir,Async_Flush,Almost_Full,Almost_Empty,PUSH_FLAG,POP_
↪FLAG,DOUT);
    input Fifo_Push_Flush,Fifo_Pop_Flush;
    input Push_Clk,Pop_Clk;
    input PUSH,POP;
    input [15:0] DIN;
    input Push_Clk_En,Pop_Clk_En,Fifo_Dir,Async_Flush;
    output [15:0] DOUT;
    output [3:0] PUSH_FLAG,POP_FLAG;
    output Almost_Full,Almost_Empty;

    parameter data_depth_int = 512;
    parameter data_width_int = 16;
    parameter reg_rd_int = 0;
    parameter sync_fifo_int = 0;

    FIFO_8K_BLK # (.data_depth_int(data_depth_int),.data_width_int(data_width_int),.reg_
↪rd_int(reg_rd_int),.sync_fifo_int(sync_fifo_int)
    )
    FIFO_INST ( .DIN(DIN), .PUSH(PUSH), .POP(POP), .Fifo_Push_Flush(Fifo_Push_Flush),
↪.Fifo_Pop_Flush(Fifo_Pop_Flush), .Push_Clk(Push_Clk),
    .Pop_Clk(Pop_Clk),.PUSH_FLAG(PUSH_FLAG), .POP_FLAG(POP_FLAG), .Push_Clk_
↪En(Push_Clk_En), .Pop_Clk_En(Pop_Clk_En),

```

(continues on next page)

(continued from previous page)

```

        .Fifo_Dir(Fifo_Dir), .Async_Flush(Async_Flush), .Almost_Full(Almost_Full), .
        ↪Almost_Empty(Almost_Empty), .DOUT(DOUT));
    endmodule

```

## Example 2: Horizontal Concatenation

- FIFO 512 x 32
- Asynchronous FIFO

```

module af512x32_512x32 (DIN,Fifo_Push_Flush,Fifo_Pop_Flush,PUSH,POP,Push_Clk,Pop_Clk,
    ↪Push_Clk_En,Pop_Clk_En,Fifo_Dir,Async_Flush,Almost_Full,Almost_Empty,PUSH_FLAG,POP_
    ↪FLAG,DOUT);
    input Fifo_Push_Flush,Fifo_Pop_Flush;
    input Push_Clk,Pop_Clk;
    input PUSH,POP;
    input [31:0] DIN;
    input Push_Clk_En,Pop_Clk_En,Fifo_Dir,Async_Flush;
    output [31:0] DOUT;
    output [3:0] PUSH_FLAG,POP_FLAG;
    output Almost_Full,Almost_Empty;

    parameter data_depth_int = 512;
    parameter data_width_int = 32;
    parameter reg_rd_int = 0;
    parameter sync_fifo_int = 0;

    FIFO_16K_BLK # (.data_depth_int(data_depth_int),.data_width_int(data_width_int),.
    ↪reg_rd_int(reg_rd_int),.sync_fifo_int(sync_fifo_int)
        )
        FIFO_INST ( .DIN(DIN), .PUSH(PUSH), .POP(POP), .Fifo_Push_Flush(Fifo_Push_Flush),
    ↪.Fifo_Pop_Flush(Fifo_Pop_Flush), .Push_Clk(Push_Clk),
            .Pop_Clk(Pop_Clk), .PUSH_FLAG(PUSH_FLAG), .POP_FLAG(POP_FLAG), .Push_Clk_
    ↪En(Push_Clk_En), .Pop_Clk_En(Pop_Clk_En), .Fifo_Dir(Fifo_Dir),
            .Async_Flush(Async_Flush), .Almost_Full(Almost_Full), .Almost_Empty(Almost_
    ↪Empty), .DOUT(DOUT));
endmodule

```

## Example 3: Vertical Concatenation

- FIFO 1024 x 16
- Synchronous FIFO

```

module f1024x16_1024x16 (DIN,Fifo_Push_Flush,Fifo_Pop_Flush,PUSH,POP,Clock,Clk_En,Fifo_Dir,
    ↪Async_Flush,Almost_Full,Almost_Empty,PUSH_FLAG,POP_FLAG,DOUT);
    input Fifo_Push_Flush,Fifo_Pop_Flush;
    input Clk;
    input PUSH,POP;
    input [15:0] DIN;
    input Clk_En,Fifo_Dir,Async_Flush;
    output [15:0] DOUT;
    output [3:0] PUSH_FLAG,POP_FLAG;

```

(continues on next page)

(continued from previous page)

```

output Almost_Full,Almost_Empty;

parameter data_depth_int = 1024;
parameter data_width_int = 16;
parameter reg_rd_int = 0;
parameter sync_fifo_int = 1;

FIFO_16K_BLK # (.data_depth_int(data_depth_int),.data_width_int(data_width_int),.
↪reg_rd_int(reg_rd_int),.sync_fifo_int(sync_fifo_int)
)
FIFO_INST (.DIN(DIN), .PUSH(PUSH), .POP(POP), .Fifo_Push_Flush(Fifo_Push_Flush), .
↪Fifo_Pop_Flush(Fifo_Pop_Flush), .Push_Clk(Clk),
        .Pop_Clk(Clk), .PUSH_FLAG(PUSH_FLAG), .POP_FLAG(POP_FLAG), .Push_Clk_En(Clk_
↪En), .Pop_Clk_En(Clk_En), .Fifo_Dir(Fifo_Dir),
        .Async_Flush(Async_Flush), .Almost_Full(Almost_Full), .Almost_Empty(Almost_
↪Empty), .DOUT(DOUT));
endmodule

```

## Multiplier Usage

The Multiplier macros are present at: <Install\_Path>/conda/share/yosys/quicklogic

### 1. 32 x 32 Multiplier

- Configured as 1 32x32 multiplier

```

module MULT_32BIT (Amult, Bmult, Valid_mult, Cmult);
input [31:0] Amult;
input [31:0] Bmult;
input Valid_mult;
output [63:0] Cmult;

wire [1:0] valit_int;

assign valit_int = {Valid_mult,Valid_mult};

//qlal4s3_mult_cell_macro
qlal4s3_mult_cell_macro u_qlal4s3_mult_cell_macro (.Amult(Amult), .Bmult(Bmult), .
↪Valid_mult(valit_int), .sel_mul_32x32(1'b1), .Cmult(Cmult));
endmodule

```

### 2. 16 x 16 Multiplier

- Configured as 1 16x16 multiplier

```

module MULT_16BIT_X2 ( Amult1, Bmult1, Valid_mult1, Cmult1,
                      Amult2, Bmult2, Valid_mult2, Cmult2 );
input [15:0] Amult1;
input [15:0] Bmult1;
input Valid_mult1;
output [31:0] Cmult1;

input [15:0] Amult2;

```

(continues on next page)



(continued from previous page)

```

input [15:0] Bmult2;
input Valid_mult2;
output [31:0] Cmult2;

wire [31:0] amult_int;
wire [31:0] bmult_int;
wire [63:0] cmult_int;
wire [1:0] valit_int;

assign valit_int = {Valid_mult2,Valid_mult1};
assign amult_int = {Amult2,Amult1};
assign bmult_int = {Bmult2,Bmult1};
assign Cmult1 = cmult_int[31:0];
assign Cmult2 = cmult_int[63:32];

//qlal4s3_mult_cell_macro
qlal4s3_mult_cell_macro u_qlal4s3_mult_cell_macro (.Amult(amult_int), .Bmult(bmult_
↪int), .Valid_mult(valit_int), .sel_mul_32x32(1'b0), .Cmult(cmult_int));
endmodule

```

### Design example Using SRAMs

The Design example using S3 SRAM block are present at: <Install\_Path>/quicklogic-arch-defs/tests/ram\_test

Address Map:

Register	Description	Reset Value	Remarks
0x40020000	FPGA IP ID	0x56A37E57	Read only
0x40020004	Revision Number	0x100	Read only (Rev. 1.00)
0x40020008	GPIO Input	0x0	GPIO_IN [7:0], Input to FPGA IP from External PAD
0x4002000C	GPIO Output	0x0	GPIO_OUT [7:0], Output from FPGA IP to External PAD
0x40020010	GPIO Direction Control	0x0	GPIO_OE [7:0] 0 – Tri-State Output 1 – Drive Output
0x40022000 – 0x400227FC	RAM 0 Access port (512x16_512x16)		RAM Input and Output Port (Write and Read from the Wishbone Interface)
0x40024000 – 0x40024FFC	RAM 1 Access port (1024x16_1024x16)		RAM Input and Output Port (Write and Read from the Wishbone Interface)
0x40026000 – 0x40026FFC	RAM 2 Access port (1024x8_1024x8)		RAM Input and Output Port (Write and Read from the Wishbone Interface)
0x40028000 – 0x400287FC	RAM 3 Access port (512x32_512x32)		RAM Input and Output Port (Write and Read from the Wishbone Interface)
0x4002a000 – 0x4002bFFC	RAM 4 Access port (2048x8_2048x8)		RAM Input and Output Port (Write and Read from the Wishbone Interface)
0x40028000 – 0x4003FFFC	Not Used	0x0	

RAM Initialization files are at: *<Install\_Path>/quicklogic-arch-defs/tests/ram\_test* Ex. *init\_2048x8.hex*, *init\_512x32.hex*, *init\_1024x8.hex*

### Design example Using FIFOs

The Design example using S3 FIFO block are present at: *<Install\_Path>/quicklogic-arch-defs/tests/fifo\_test*

Address Map:

Register	Description	Reset Value	Remarks
0x40020000	FPGA IP ID	0xF1F07E57	Read only
0x40020004	Revision Number	0x100	Read only (Rev. 1.00)
0x40020008	GPIO Input	0x0	GPIO_IN [7:0], Input to FPGA IP from External PAD
0x4002000C	GPIO Output	0x0	GPIO_OUT [7:0], Output from FPGA IP to External PAD
0x40020010	GPIO Direction Control	0x0	GPIO_OE [7:0] 0 – Tri-State Output 1 – Drive Output
0x40020100	FIFO 1 Access port (af512x16_512x16)		Asynchronous FIFO FIFO Push & Pop port
0x40020104	FIFO 1 Flags		bit [31:16] – Reserved <b>bit [15] - Almost_Empty</b> bit [14:12] – Reserved <b>bit [11:8] – POP Flag</b> <b>bit [7] - Almost_Full</b> bit [6:4] – Reserved <b>bit [3:0] – PUSH Flag</b>
0x40020200	FIFO 2 Access port (f1024x16_1024x16)		Synchronous FIFO FIFO Push & Pop port
0x40020204	FIFO 2 Flags		bit [31:16] – Reserved <b>bit [15] - Almost_Empty</b> bit [14:12] – Reserved <b>bit [11:8] – POP Flag</b> <b>bit [7] - Almost_Full</b> bit [6:4] – Reserved <b>bit [3:0] – PUSH Flag</b>
0x40020400	FIFO 3 Access port (af512x32_512x32)		Asynchronous FIFO FIFO Push & Pop port
0x40020404	FIFO 3 Flags		bit [31:16] – Reserved <b>bit [15] - Almost_Empty</b> bit [14:12] – Reserved <b>bit [11:8] – POP Flag</b> <b>bit [7] - Almost_Full</b> bit [6:4] – Reserved <b>bit [3:0] – PUSH Flag</b>
0x40020800 – 0x4003FFFC	Not Used	0x0	

Push Flag Description:

Value	Status
0000	Full
0001	Empty
0010	Room for more than one-half
0011	Room for more than one-fourth
0100	Room for less than one-fourth full to 64
1010	Room for 32 to 63
1011	Room for 16 to 31
1100	Room for 8 to 15
1101	Room for 4 to 7
1110	Room for at least 2
1111	Room for at least 1
Others	Reserved

Pop Flag Description:

Value	Status
0000	Empty
0001	1 entry in FIFO
0010	At least 2 entries in FIFO
0011	At least 4 entries in FIFO
0100	At least 8 entries in FIFO
0101	At least 16 entries in FIFO
0110	At least 32 entries in FIFO
1000	Less than one-fourth to 64 full
1101	One-fourth or more full
1110	One-half or more full
1111	Full
Others	Reserved

## Symbols

24x24 Device Pin Mapping CSV file, 18

## B

BRAM and DSP Inference, 37

## D

Design example Using FIFOs:, 62

Design example Using SRAMs:, 61

Design flow, 5

## F

FIFO Features:, 55

FIFO Usage:, 58

## G

Generate the ASCII header file format, 47

Generate the Binary file format, 48

Generate the Jlink and OpenOCD file, 47

Generate the Programming files, 17

## H

Hardware Limitations and Online References,  
54

## I

Installing Symbiflow on Linux, 1

## M

Macro Usage and examples:, 56

Multiplier Features:, 55

Multiplier Usage:, 60

## O

Online References, 40

## P

PCF sample, 18, 53

Performing Design Synthesis, 7, 44

Performing the Post-Layout Simulation (*verify-  
the post layout verilog file*), 17

Performing the Post-Layout Simulation (*verify-  
ing the configuration bits*), 46

Performing the Pre-Layout Simulation, 6, 43

Pin Mapping, 8

## R

RAM Features:, 55

RAM Usage:, 56

Run design flow on a simple counter design  
(*eos-s3*), 43

Running Pack, Place and Route Tools, 16

Running pack, Place and Route tools (*eos-s3*,  
45

## S

S3B Device:, 55

Supported Commands, 4, 42

Symbiflow: Design flow (*eos-s3*), 41

Symbiflow: Design flow (*qlf\_k4n8/qlf\_k6n10*), 3