S3B Hardmacro (RAM/FIFO /Multiplier) User Guide

About This Document

This document provides the details of in-built RAMs/FIFOs and Multipliers in S3B and their usage in the user design

S3B Device Hardmacro Resources:

1. RAMs:
    - 8 blocks of 8K bits
2. FIFO:
    - 8 in-build FIFO controllers
    - Can be configured into FIFOs using above RAM blocks
3. Multipliers
    - 2 32X32 in-built multipliers

RAM Features:

- 8Kbits/RAM
- x9, x18 and x36 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

FIFO Features:

- x9, x18 and x36 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

Multiplier Features:

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

RAM Usage:

The RAM macro (RAM_16K_BLK.v) is present at:
<Install_Path>/antmicro_install/share/yosys/quicklogic/

The examples of RAM usage are present at:
<Install_Path> /symbiflow-arch-defs/quicklogic /pp3/tests/RAM_Examples

1. RAM macro is a 16K bits block which can be configured as:
    - 2 independent 8Kbits RAM blocks
    - A horizontal or vertical concatenated block (16Kbits)
2. Concatenation_En = 0, then RAM can be configured into 2 8Kbits block
3. Concatenation_En = 1, then RAM can be configured into 1 16Kbits block

Example1 (2 8Kbits RAM):

- 2 independent RAMs of 512x16 (8Kbits each)
- Concatenation_En = 0

```
module r512x16_512x16_r1024x8_1024x8 (WA0,RA0,WD0,WD0_SEL,RD0_SEL,WClk0,RClk0,WClk0_En,RClk0_En,WEN0,RD0,
                                      WA1,RA1,WD1,WD1_SEL,RD1_SEL,WClk1,RClk1,WClk1_En,RClk1_En,WEN1,RD1
                                      );

input [8:0] WA0;
input [8:0] RA0;
input WD0_SEL,RD0_SEL;
input WClk0,RClk0;
input WClk0_En,RClk0_En;
input [1:0] WEN0;
input [15:0] WD0;
output [15:0] RD0;

input [9:0] WA1;
input [9:0] RA1;
input WD1_SEL,RD1_SEL;
input WClk1,RClk1;
input WClk1_En,RClk1_En;
input WEN1;
input [7:0] WD1;
output [7:0] RD1;

parameter Concatenation_En = 0 ;

parameter wr_addr_int0 = 9 ;
parameter rd_addr_int0 = 9;
parameter wr_depth_int0 = 512;
parameter rd_depth_int0 = 512;
parameter wr_width_int0 = 16;
parameter rd_width_int0 = 16;
parameter wr_enable_int0 = 2;
parameter reg_rd_int0 = 0;

parameter wr_addr_int1 = 10;
parameter rd_addr_int1 = 10;
parameter wr_depth_int1 = 1024;
parameter rd_depth_int1 = 1024;
parameter wr_width_int1 = 8;
parameter rd_width_int1 = 8;
parameter wr_enable_int1 = 1;
parameter reg_rd_int1 = 0;
```

```
RAM_16K_BLK #(Concatenation_En,
              wr_addr_int0,rd_addr_int0,wr_depth_int0,rd_depth_int0,wr_width_int0,rd_width_int0,wr_enable_int0,reg_rd_int0,
              wr_addr_int1,rd_addr_int1,wr_depth_int1,rd_depth_int1,wr_width_int1,rd_width_int1,wr_enable_int1,reg_rd_int1
              )
RAM_INST (  .WA0(WA0),
            .RA0(RA0),
            .WD0(WD0),
            .WD0_SEL(WD0_SEL),
            .RD0_SEL(RD0_SEL),
            .WClk0(WClk0),
            .RClk0(RClk0),
            .WClk0_En(WClk0_En),
            .RClk0_En(RClk0_En),
            .WEN0(WEN0),
            .RD0(RD0),

            .WA1(WA1),
            .RA1(RA1),
            .WD1(WD1),
            .WD1_SEL(WD1_SEL),
            .RD1_SEL(RD1_SEL),
            .WClk1(WClk1),
            .RClk1(RClk1),
            .WClk1_En(WClk1_En),
            .RClk1_En(RClk1_En),
            .WEN1(WEN1),
            .RD1(RD1),

            .LS(1'b0),
            .SD(1'b0),
            .DS(1'b0),
            .LS_RB1(1'b0),
            .SD_RB1(1'b0),
            .DS_RB1(1'b0)
           );
endmodule
```

Example 2 (Horizontal Concatenation):

- 512x32 RAM (16 Kbits)
- Concatenation_En = 1

```
module r512x32_512x32 (WA,RA,WD,WD_SEL,RD_SEL,WClk,RClk,WClk_En,RClk_En,WEN,RD);

input [8:0] WA;
input [8:0] RA;
input WD_SEL,RD_SEL;
input WClk,RClk;
input WClk_En,RClk_En;
input [3:0] WEN;
input [31:0] WD;
output [31:0] RD;

parameter Concatenation_En = 1 ;

parameter wr_addr_int0 = 9 ;
parameter rd_addr_int0 = 9;
parameter wr_depth_int0 = 512;
parameter rd_depth_int0 = 512;
parameter wr_width_int0 = 32;
parameter rd_width_int0 = 32;
parameter wr_enable_int0 = 4;
parameter reg_rd_int0 = 0;

RAM_16K_BLK #(Concatenation_En,
              wr_addr_int0,rd_addr_int0,wr_depth_int0,rd_depth_int0,wr_width_int0,rd_width_int0,wr_enable_int0,reg_rd_int0
              )
RAM_INST (  .WA0(WA),
            .RA0(RA),
            .WD0(WD),
            .WD0_SEL(WD_SEL),
            .RD0_SEL(RD_SEL),
            .WClk0(WClk),
            .RClk0(RClk),
            .WClk0_En(WClk_En),
            .RClk0_En(RClk_En),
            .WEN0(WEN),
            .RD0(RD),

            .LS(1'b0),
            .SD(1'b0),
            .DS(1'b0),
            .LS_RB1(1'b0),
            .SD_RB1(1'b0),
            .DS_RB1(1'b0)
            );

endmodule
```

Example 3 (Vertical Concatenation):

- 2048 x 8 RAM (16 Kbits)
- Concatenation_En = 1

```
module r2048x8_2048x8 (WA,RA,WD,WD_SEL,RD_SEL,WClk,RClk,WClk_En,RClk_En,WEN,RD);

input [10:0] WA;
input [10:0] RA;
input WD_SEL,RD_SEL;
input WClk,RClk;
input WClk_En,RClk_En;
input WEN;
input [7:0] WD;
output [7:0] RD;

parameter Concatenation_En = 1 ;

parameter wr_addr_int0 = 11 ;
parameter rd_addr_int0 = 11;
parameter wr_depth_int0 = 2048;
parameter rd_depth_int0 = 2048;
parameter wr_width_int0 = 8;
parameter rd_width_int0 = 8;
parameter wr_enable_int0 = 1;
parameter reg_rd_int0 = 0;

RAM_16K_BLK #(Concatenation_En,
              wr_addr_int0,rd_addr_int0,wr_depth_int0,rd_depth_int0,wr_width_int0,rd_width_int0,wr_enable_int0,reg_rd_int0
              )
RAM_INST (  .WA0(WA),
            .RA0(RA),
            .WD0(WD),
            .WD0_SEL(WD_SEL),
            .RD0_SEL(RD_SEL),
            .WClk0(WClk),
            .RClk0(RClk),
            .WClk0_En(WClk_En),
            .RClk0_En(RClk_En),
            .WEN0(WEN),
            .RD0(RD),

            .LS(1'b0),
            .SD(1'b0),
            .DS(1'b0),
            .LS_RB1(1'b0),
            .SD_RB1(1'b0),
            .DS_RB1(1'b0)
            );

endmodule
```

FIFO Usage:

The FIFO macro (FIFO_16K_BLK.v) is present at:
<Install_Path>/antmicro_install/share/yosys/quicklogic/

The examples of FIFO usage are present at:
<Install_Path> /symbiflow-arch-defs/quicklogic/ pp3/tests/FIFO_Examples

1. FIFO macro is a 16K bits block which can be configured as:
    - 2 independent 8Kbits FIFO blocks
    - A horizontal or vertical concatenated block (16Kbits)
2. Concatenation_En = 0, then FIFO can be configured into 2 8Kbits block
3. Concatenation_En = 1, then FIFO can be configured into 1 16Kbits block

Example 1 (2 8Kbits FIFOs):

- 2 independent FIFOs of 512x16
    - One asynchronous FIFO of af512x16 (sync_fifo_int0 = 0)
    - One synchronous FIFO of f512x16 (sync_fifo_int1 = 1)
- Concatenation_En = 0

```
module af512x16_512x16_f512x16_512x16 (
                DIN0,Fifo_Push_Flush0,Fifo_Pop_Flush0,PUSH0,POP0,Push_Clk0,Pop_Clk0,Push_Clk0_En,Pop_Clk0_En,Fifo0_Dir, Async_Flush0,Almost_Full0,Almost_Empty0,PUSH_FLAG0,POP_FLAG0,DOUT0,
                DIN1,Fifo_Push_Flush1,Fifo_Pop_Flush1,PUSH1,POP1,Clk1,Clk1_En,Fifo1_Dir,Async_Flush1,Almost_Full1,Almost_Empty1,PUSH_FLAG1,POP_FLAG1,DOUT1
                                );

input Fifo_Push_Flush0,Fifo_Pop_Flush0;
input Push_Clk0,Pop_Clk0;
input PUSH0,POP0;
input [15:0] DIN0;
input Push_Clk0_En,Pop_Clk0_En,Fifo0_Dir,Async_Flush0;
output [15:0] DOUT0;
output [3:0] PUSH_FLAG0,POP_FLAG0;
output Almost_Full0,Almost_Empty0;

input Fifo_Push_Flush1,Fifo_Pop_Flush1;
input Clk1;
input PUSH1,POP1;
input [15:0] DIN1;
input Clk1_En,Fifo1_Dir,Async_Flush1;
output [15:0] DOUT1;
output [3:0] PUSH_FLAG1,POP_FLAG1;
output Almost_Full1,Almost_Empty1;

parameter Concatenation_En = 0;

parameter wr_depth_int0 = 512;
parameter rd_depth_int0 = 512;
parameter wr_width_int0 = 16;
parameter rd_width_int0 = 16;
parameter reg_rd_int0 = 0;
parameter sync_fifo_int0 = 0;

parameter wr_depth_int1 = 512;
parameter rd_depth_int1 = 512;
parameter wr_width_int1 = 16;
parameter rd_width_int1 = 16;
parameter reg_rd_int1 = 0;
parameter sync_fifo_int1 = 1;
```

```
FIFO_16K_BLK   #(Concatenation_En,
                wr_depth_int0,rd_depth_int0,wr_width_int0,rd_width_int0,reg_rd_int0,sync_fifo_int0,
                wr_depth_int1,rd_depth_int1,wr_width_int1,rd_width_int1,reg_rd_int1,sync_fifo_int1
                )
FIFO_INST      (.DIN0(DIN0),
                .PUSH0(PUSH0),
                .POP0(POP0),
                .Fifo_Push_Flush0(Fifo_Push_Flush0),
                .Fifo_Pop_Flush0(Fifo_Pop_Flush0),
                .Push_Clk0(Push_Clk0),
                .Pop_Clk0(Pop_Clk0),
                .PUSH_FLAG0(PUSH_FLAG0),
                .POP_FLAG0(POP_FLAG0),
                .Push_Clk0_En(Push_Clk0_En),
                .Pop_Clk0_En(Pop_Clk0_En),
                .Fifo0_Dir(Fifo0_Dir),
                .Async_Flush0(Async_Flush0),
                .Almost_Full0(Almost_Full0),
                .Almost_Empty0(Almost_Empty0),
                .DOUT0(DOUT0),

                .DIN1(DIN1),
                .PUSH1(PUSH1),
                .POP1(POP1),
                .Fifo_Push_Flush1(Fifo_Push_Flush1),
                .Fifo_Pop_Flush1(Fifo_Pop_Flush1),
                .Push_Clk1(Clk1),
                .Pop_Clk1(Clk1),
                .PUSH_FLAG1(PUSH_FLAG1),
                .POP_FLAG1(POP_FLAG1),
                .Push_Clk1_En(Clk1_En),
                .Pop_Clk1_En(Clk1_En),
                .Fifo1_Dir(Fifo1_Dir),
                .Async_Flush1(Async_Flush1),
                .Almost_Full1(Almost_Full1),
                .Almost_Empty1(Almost_Empty1),
                .DOUT1(DOUT1),

                .LS(1'b0),
                .SD(1'b0),
                .DS(1'b0),
                .LS_RB1(1'b0),
                .SD_RB1(1'b0),
                .DS_RB1(1'b0)
                );

endmodule
```

Example 2 (Horizontal Concatenation):

- One asynchronous FIFO af512x32 (16 Kbits)
    - asynchronous FIFO of 512x32 (sync_fifo_int0 = 0)
- Concatenation_En = 1

```
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 Concatenation_En = 1;

parameter wr_depth_int0 = 512;
parameter rd_depth_int0 = 512;
parameter wr_width_int0 = 32;
parameter rd_width_int0 = 32;
parameter reg_rd_int0 = 0;
parameter sync_fifo_int0 = 0;


FIFO_16K_BLK   #(Concatenation_En,
                 wr_depth_int0,rd_depth_int0,wr_width_int0,rd_width_int0,reg_rd_int0,sync_fifo_int0
                 )
FIFO_INST      (.DIN0(DIN),
                .PUSH0(PUSH),
                .POP0(POP),
                .Fifo_Push_Flush0(Fifo_Push_Flush),
                .Fifo_Pop_Flush0(Fifo_Pop_Flush),
                .Push_Clk0(Push_Clk),
                .Pop_Clk0(Pop_Clk),
                .PUSH_FLAG0(PUSH_FLAG),
                .POP_FLAG0(POP_FLAG),
                .Push_Clk0_En(Push_Clk_En),
                .Pop_Clk0_En(Pop_Clk_En),
                .Fifo0_Dir(Fifo_Dir),
                .Async_Flush0(Async_Flush),
                .Almost_Full0(Almost_Full),
                .Almost_Empty0(Almost_Empty),
                .DOUT0(DOUT),

                .LS(1'b0),
                .SD(1'b0),
                .DS(1'b0),
                .LS_RB1(1'b0),
                .SD_RB1(1'b0),
                .DS_RB1(1'b0)
                );

endmodule
```

Example 3 (Vertical Concatenation):

- One asynchronous FIFO af2048x8(16 Kbits)
    - asynchronous FIFO of 2048x8 (sync_fifo_int0 = 0)
- Concatenation_En = 1

```
module af2048x8_2048x8(
       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 [7:0] DIN;
input Push_Clk_En,Pop_Clk_En,Fifo_Dir,Async_Flush;
output [7:0] DOUT;
output [3:0] PUSH_FLAG,POP_FLAG;
output Almost_Full,Almost_Empty;

parameter Concatenation_En = 1;

parameter wr_depth_int0 = 2048;
parameter rd_depth_int0 = 2048;
parameter wr_width_int0 = 8;
parameter rd_width_int0 = 8;
parameter reg_rd_int0 = 0;
parameter sync_fifo_int0 = 0;


FIFO_16K_BLK   #(Concatenation_En,
                 wr_depth_int0,rd_depth_int0,wr_width_int0,rd_width_int0,reg_rd_int0,sync_fifo_int0
                 )
FIFO_INST      (.DIN0(DIN),
                .PUSH0(PUSH),
                .POP0(POP),
                .Fifo_Push_Flush0(Fifo_Push_Flush),
                .Fifo_Pop_Flush0(Fifo_Pop_Flush),
                .Push_Clk0(Push_Clk),
                .Pop_Clk0(Pop_Clk),
                .PUSH_FLAG0(PUSH_FLAG),
                .POP_FLAG0(POP_FLAG),
                .Push_Clk0_En(Push_Clk_En),
                .Pop_Clk0_En(Pop_Clk_En),
                .Fifo0_Dir(Fifo_Dir),
                .Async_Flush0(Async_Flush),
                .Almost_Full0(Almost_Full),
                .Almost_Empty0(Almost_Empty),
                .DOUT0(DOUT),

                .LS(1'b0),
                .SD(1'b0),
                .DS(1'b0),
                .LS_RB1(1'b0),
                .SD_RB1(1'b0),
                .DS_RB1(1'b0)
                );

endmodule
```

Example 4 (Vertical Concatenation):

- One synchronous FIFO f2048x8_1024x16 (16 Kbits)
    - asynchronous FIFO (sync_fifo_int0 = 0)
    - Push side data width = 8
    - Pop side data width = 16
- Concatenation_En = 1

```
module f2048x8_1024x16(
        DIN,Fifo_Push_Flush,Fifo_Pop_Flush,PUSH,POP,Clk,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 [7:0] DIN;
input Clk_En,Fifo_Dir,Async_Flush;
output [15:0] DOUT;
output [3:0] PUSH_FLAG,POP_FLAG;
output Almost_Full,Almost_Empty;

parameter Concatenation_En = 1;

parameter wr_depth_int0 = 2048;
parameter rd_depth_int0 = 1024;
parameter wr_width_int0 = 8;
parameter rd_width_int0 = 16;
parameter reg_rd_int0 = 0;
parameter sync_fifo_int0 = 1;

FIFO_16K_BLK   #(Concatenation_En,
                 wr_depth_int0,rd_depth_int0,wr_width_int0,rd_width_int0,reg_rd_int0,sync_fifo_int0
                 )
FIFO_INST      (.DIN0(DIN),
                .PUSH0(PUSH),
                .POP0(POP),
                .Fifo_Push_Flush0(Fifo_Push_Flush),
                .Fifo_Pop_Flush0(Fifo_Pop_Flush),
                .Push_Clk0(Clk),
                .Pop_Clk0(Clk),
                .PUSH_FLAG0(PUSH_FLAG),
                .POP_FLAG0(POP_FLAG),
                .Push_Clk0_En(Clk_En),
                .Pop_Clk0_En(Clk_En),
                .Fifo0_Dir(Fifo_Dir),
                .Async_Flush0(Async_Flush),
                .Almost_Full0(Almost_Full),
                .Almost_Empty0(Almost_Empty),
                .DOUT0(DOUT),

                .LS(1'b0),
                .SD(1'b0),
                .DS(1'b0),
                .LS_RB1(1'b0),
                .SD_RB1(1'b0),
                .DS_RB1(1'b0)
                );

endmodule
```

Multiplier Usage:

The Multiplier macros are present at:
<Install_Path>/antmicro_install/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;
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> /symbiflow-arch-defs/quicklogic/ pp3/tests/quicklogic_testsuite/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
0x40021000 – 0x400217FC	RAM 1 Access port (512x16_512x16)		RAM Input and Output Port (Write and Read from the Wishbone Interface)
0x40022000 – 0x40022FFC	RAM 2 Access port (1024x8_1024x8)		RAM Input and Output Port (Write and Read from the Wishbone Interface)
0x40024000 – 0x400247FC	RAM 3 Access port (512x32_512x32)		RAM Input and Output Port (Write and Read from the Wishbone Interface)
0x40028000 – 0x4003FFFC	Not Used	0x0	

Design example Using FIFOs:

The Design example using S3 FIFO block are present at:
<Install_Path> /symbiflow-arch-defs/quicklogic/ pp3/tests/quicklogic_testsuite/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 **bit [15] - Almost_Empty** bit [14:12] – Reserved **bit [11:8] – POP Flag** **bit [7] - Almost_Full** bit [6:4] – Reserved **bit [3:0] – PUSH Flag**
0x40020200	FIFO 2 Access port (af512x16_512x16)		Synchronous FIFO FIFO Push & Pop port
0x40020204	FIFO 2 Flags		bit [31:16] – Reserved **bit [15] - Almost_Empty** bit [14:12] – Reserved **bit [11:8] – POP Flag** **bit [7] - Almost_Full** bit [6:4] – Reserved **bit [3:0] – PUSH Flag**
0x40020400	FIFO 3 Access port (af512x32_512x32)		Asynchronous FIFO FIFO Push & Pop port

0x40020404	FIFO 3 Flags		bit [31:16] – Reserved **bit [15] - Almost_Empty** bit [14:12] – Reserved **bit [11:8] – POP Flag** **bit [7] - Almost_Full** bit [6:4] – Reserved **bit [3:0] – PUSH Flag**
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-forth
0100	Room for less than one-forth 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-forth to 64 full
1101	One-forth or more full
1110	One-half or more full
1111	Full
Others	Reserved