

FPGA compile 수정 방법

- 1 : ../common/ipcore_dir/ 내에 수정하고 싶은 .vhd, .xco 파일을 확인한다 .

ex) ram_1kx15 의 경우

```
m_1kx15.asy   ram_1kx15.ngc   ram_1kx15.veo   ram_1kx15.xco  
m_1kx15.gise  ram_1kx15.sym   ram_1kx15.vhd   ram_1kx15.xise
```

- 2 : 해당 vhd 파일의 복사본을 만들어 준다 .

ex) ram_1kx15_modified.vhd 생성

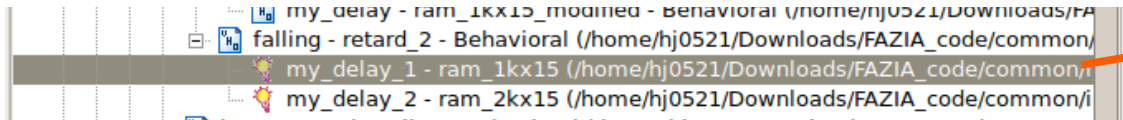
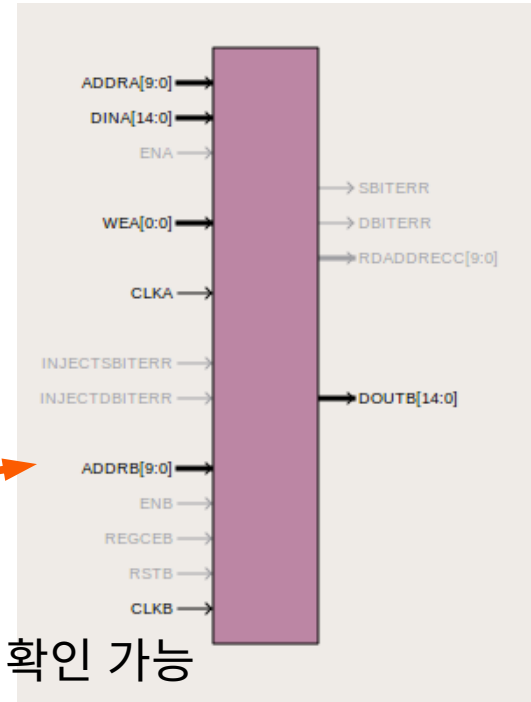
- 3 : 해당 xco 파일과 ise 를 이용하여 해당 vhd 소스를 열어 수정하려는 ram 종류와 전체적인 데이터 크기를 확인한다 .

ex) ram_1kx15. 의 경우

```

SELECT Block_Memory_Generator family Xilinx, Inc. 3.3
# END Select
# BEGIN Parameters
CSET additional_inputs_for_power_estimation=false
CSET algorithm=Fixed_Primitives
CSET assume_synchronous_clk=false
CSET byte_size=9
CSET coe_file=no_coe_file_loaded
CSET collision_warnings=ALL
CSET component_name=ram_1kx15
CSET disable_collision_warnings=false
CSET disable_out_of_range_warnings=false
CSET ecc=false
CSET enable_a=Always_Enabled
CSET enable_b=Always_Enabled
CSET error_injection_type=Single_Bit_Error_Injection
CSET fill_remaining_memory_locations=false
CSET load_init_file=false
CSET memory_type=Simple_Dual_Port_RAM
CSET operating_mode_a=READ_FIRST
CSET operating_mode_b=READ_FIRST
CSET output_reset_value_a=0
CSET output_reset_value_b=0
CSET pipeline_stages=0
CSET port_a_clock=100
CSET port_a_enable_rate=100
CSET port_a_write_rate=50
CSET port_b_clock=100
CSET port_b_enable_rate=100
CSET port_b_write_rate=0
CSET primitive=1kx18
CSET read_width_a=15
CSET read_width_b=15
CSET register_porta_output_of_memory_core=false
CSET register_porta_output_of_memory_primitives=false
CSET register_portb_output_of_memory_core=false
  
```

→ .XCO 파일을 열면 정보를 확인 할 수 있음 .



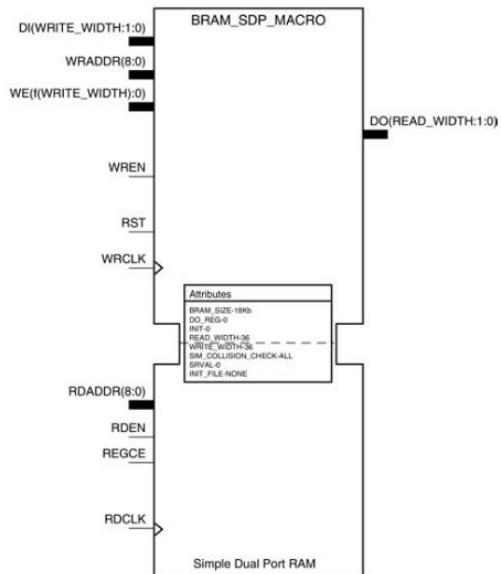
→ 혹은 아예 ise 에서 .vhd 를 열어서 내부의 데이터 크기를 가시적으로 확인 가능

- 4 : 확인한 ram 정보를 기반으로 virtex5_hdl.pdf (구글 검색시 다운로드 가능) 에 있는 적절한 port 를 채택 .

ex) ram_1kx15 의 경우 simple dual port 사용 (주로 BRAM_~ 형태의 ram 채택)

BRAM_SDP_MACRO

Macro: Simple Dual Port RAM



- 5 : 3 번에서 확인한 데이터 크기를 기초로 하여 해당 포트에 맞는 주소 크기 확인

Configuration Table

DATA_WIDTH	BRAM_SIZE	ADDR	WE
72 - 37	36kb	9	8
36 - 19	36kb	10	4
	18kb	9	
18 - 10	36kb	11	2
	18kb	10	
9 - 5	36kb	12	1
	18kb	11	
4 - 3	36kb	13	1
	18kb	12	
2	36kb	14	1
	18kb	13	
1	36kb	15	1
	18kb	14	

ram_1kx15 의 경우 해당
목록 참고

- * 모든 데이터 크기가 동일 할 수는 없기에 , 적절하게 데이터 크기 조절이 필요함

▪ 6 : ram_1kx15.vhd → ram_1kx15_modified.vhd 의 경우

```
-- THIS COPYRIGHT NOTICE AND DISCLAIMER MUST BE RETAINED AS --
-- PART OF THIS FILE AT ALL TIMES. --
-----
-- You must compile the wrapper file ram_1kx15.vhd when simulating
-- the core, ram_1kx15. When compiling the wrapper file, be sure to
-- reference the XilinxCoreLib VHDL simulation library. For detailed
-- instructions, please refer to the "CORE Generator Help".

-- The synthesis directives "translate_off/translate_on" specified
-- below are supported by Xilinx, Mentor Graphics and Synplicity
-- synthesis tools. Ensure they are correct for your synthesis tool(s).

LIBRARY ieee;
USE ieee.std_logic_1164.ALL;
-- synthesis translate_off
Library XilinxCoreLib;
-- synthesis translate_on
ENTITY ram_1kx15 IS
    port (
        clka: in std_logic;
        wea: in std_logic_vector(0 downto 0);
        addra: in std_logic_vector(9 downto 0);
        dina: in std_logic_vector(14 downto 0);
        clkb: in std_logic;
        addrb: in std_logic_vector(9 downto 0);
        doutb: out std_logic_vector(14 downto 0));
END ram_1kx15;

ARCHITECTURE ram_1kx15_a OF ram_1kx15 IS
-- synthesis translate_off
component wrapped_ram_1kx15
    port (
        clka: in std_logic;
        wea: in std_logic_vector(0 downto 0);
        addra: in std_logic_vector(9 downto 0);
        dina: in std_logic_vector(14 downto 0);
        clkb: in std_logic;
        addrb: in std_logic_vector(9 downto 0);
        doutb: out std_logic_vector(14 downto 0));
end component;
```



```
LIBRARY UNIMACRO;
LIBRARY ieee;
use unimacro.Vcomponents.all;
USE ieee.std_logic_1164.ALL;
USE IEEE.STD_LOGIC_ARITH.ALL;
USE IEEE.STD_LOGIC_UNSIGNED.ALL;

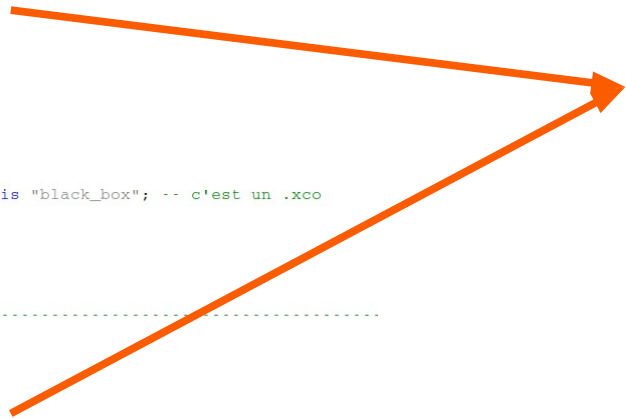
-- synthesis translate_off
Library XilinxCoreLib;
-- synthesis translate_on
ENTITY ram_1kx15_modified IS
    port (
        WRCLK: in std_logic;
        WE: in std_logic_vector(1 downto 0);
        WRADDR: in std_logic_vector(9 downto 0);
        DI: in std_logic_vector(14 downto 0);
        RDCLK: in std_logic;
        RDADDR: in std_logic_vector(9 downto 0);
        DO: out std_logic_vector(14 downto 0));
END ram_1kx15_modified;

ARCHITECTURE Behavioral OF ram_1kx15_modified IS
-- Xilinx HDL Libraries Guide, version 11.2
```

▪ 7 : core generator 의 코드 수정 : ram_1kx15.vhd → ram_1kx15_modified.vhd 로 적용

```
blockTrig - ETrigger - Behavioral (/home/hj0521/Downloads/FAZIA_code/common/e_trigger.vhd)
formeur - shaper - Behavioral (/home/hj0521/Downloads/FAZIA_code/common/shaper.vhd)
rising - retard_1 - Behavioral (/home/hj0521/Downloads/FAZIA_code/common/retard_1.vhd)
my_delay - ram_1kx15_modified - Behavioral (/home/hj0521/Downloads/FAZIA_code/common/ipcore_
```

```
43 architecture Behavioral of retard_1 is
44
45 component ram_1kx15_modified
46 port (
47   WRCLK : in std_logic;
48   WE    : in std_logic_vector( 1 downto 0);
49   WRADDR : in std_logic_vector( 9 downto 0);
50   DI    : in std_logic_vector(14 downto 0);
51   RDCLK : in std_logic;
52   RDADDR : in std_logic_vector( 9 downto 0);
53   DO    : out std_logic_vector(14 downto 0);
54 end component;
55
56 signal WRADDR    : std_logic_vector(9 downto 0);
57 signal RDADDR    : std_logic_vector(9 downto 0);
58 signal compteur  : std_logic_vector(9 downto 0);
59
60 attribute box_type : string;
61 attribute box_type of ram_1kx15_modified : component is "black_box"; -- c'est un .xco
62
63 begin
64   WRADDR <= compteur;
65   RDADDR <= unsigned(compteur) - unsigned(duree);
66
67 -----
68 make_mem: if DATA_WIDTH = 15 generate
69 my_delay : ram_1kx15_modified
70 port map (
71   WRCLK => clk,
72   WE    => "01",
73   WRADDR => WRADDR,
74   DI    => din,
75   RDCLK => clk,
76   RDADDR => RDADDR,
77   DO    => dout
78 );
79 end generate;
80
81 -----
```

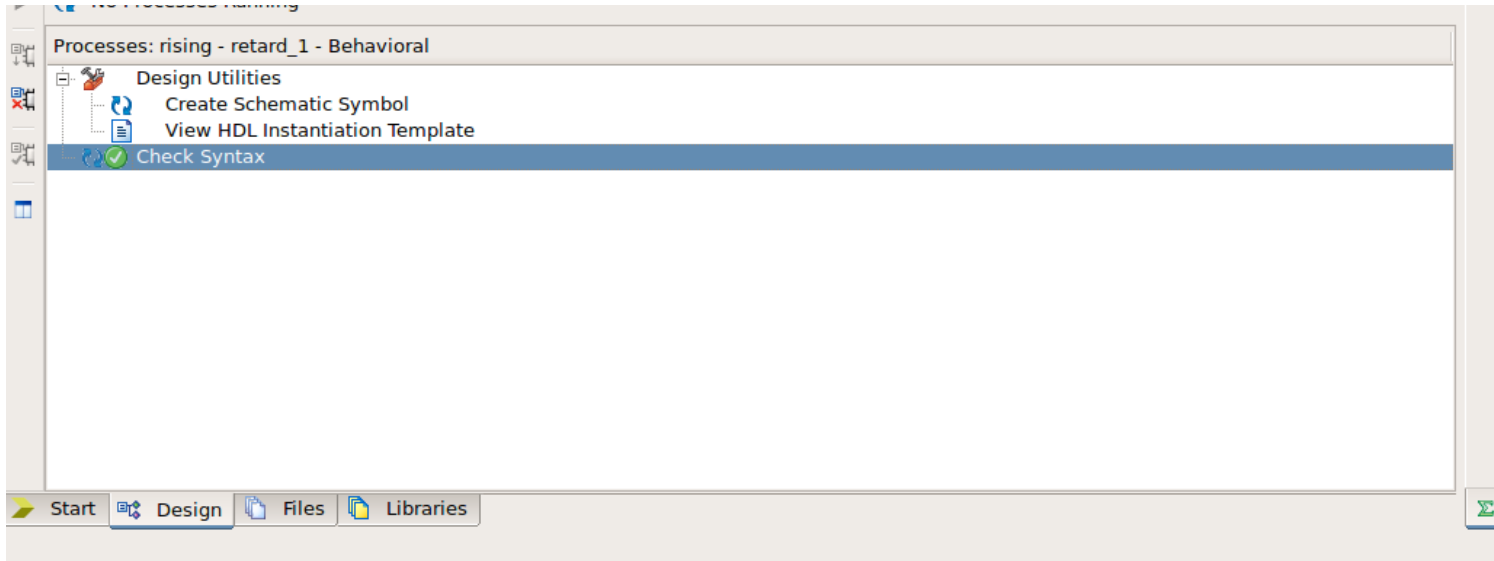


port 이름 수정

- 8 : add source 를 하면 수정된 .vhd 코드가 수정한 core generator 아래에 하위로 들어있음 .
(ise 를 종료 후 재시작을 해야 할 수도 있음)

rising - retard_1 - Behavioral (/home/hj0521/Downloads/FAZIA_code/common/retard_1.vhd)
my_delay - ram_1kx15_modified - Behavioral (/home/hj0521/Downloads/FAZIA_code/common/ipcore_dir/ran

→ 이후 Check Syntax 를 오류가 있는 부분이 있으면 수정할 것



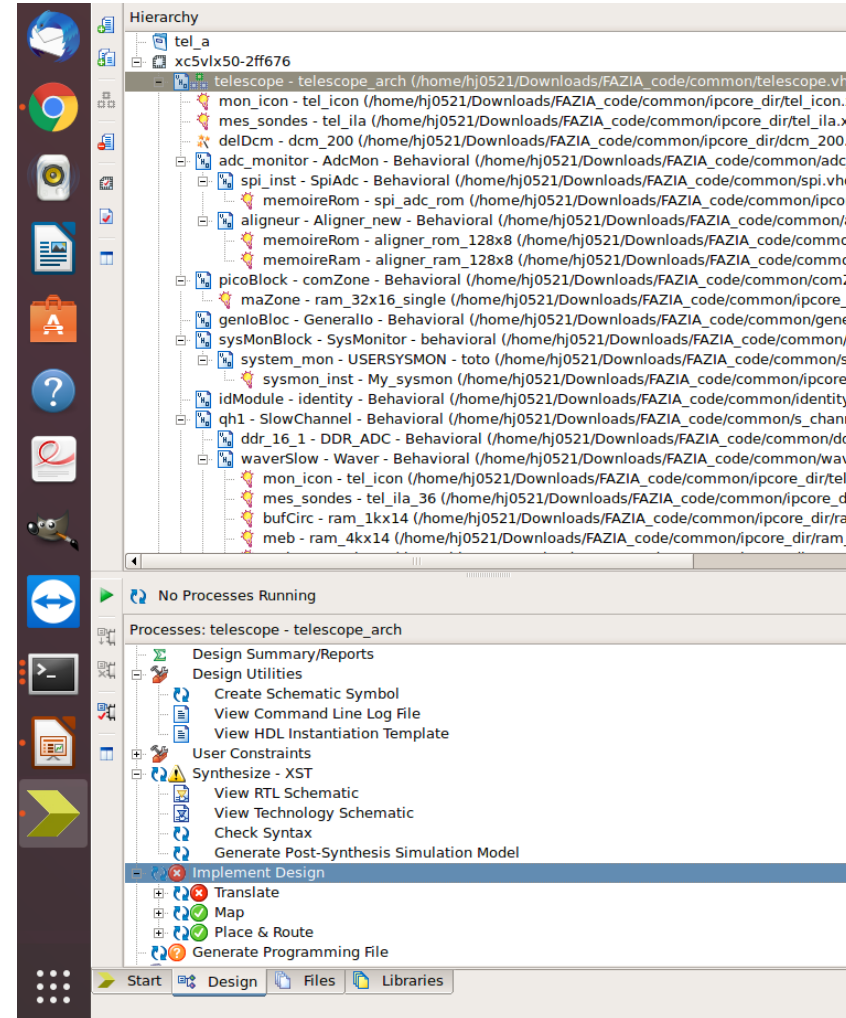
- 9 : ../tel_a/telescope.prj 에 들어가서 vhdl work “../common/ipcore_dir/(원래 core gen.).xco” 를 vhdl work “../common/ipcore_dir/(수정한 vhd 파일).vhd” 로 수정
- 10 : common/ipcore_dir(원래 xco file 있던 곳) 의 수정 전 ram_~.ngc 을 수정 후 ram_~_modified.ngc 으로 이름을 바꿈
- 11 : tel_a/ 해당 coregenerator 의 이름의 ~.prj 에 들어가 vhdl work “../common/ipcore_dir/ram_~_modified.vhd(수정한 것)” 를 추가
- 12 : tel_a.xise 를 vim 로 열어 수정하려는 ram_~ 의 이름을 ~_modified.vhd 로 수정

```

</file>
</file>
<file xil_pn:name="../common/ipcore_dir/ram_1kx15_modified.vhd" xil_pn:type="FILE_VHDL">
  <association xil_pn:name="BehavioralSimulation" xil_pn:seqID="332"/>
  <association xil_pn:name="Implementation" xil_pn:seqID="1"/>
</file>
<file xil_pn:name="ipcore_dir/get_icon.xise" xil_pn:type="FILE_COREGENISE">
  <association xil_pn:name="Implementation" xil_pn:seqID="0"/>
</file>

```

- 13 : 이후 Implement Design 을 클릭 후 모두 문제가 없으면 컴파일 성공 !



21/01/22

- 14 : syntax error 가 없이 계속 compile 이 안되는 경우 , 아래의 설정을 변경 해 볼 것 .

```
hjo521@hjo521:~$ sudo vi /opt/Xilinx/14.3/ISE_DS/ISE/bin/lin64/unwrapped/inserte
```

Xilinx 설치 경로의 ../14.3/ISE_DS/ISE/bin/lin64/unwrapped/inserte 파일을 수정 해야함 .

```
File Edit View Search Terminal Help
~/bin/sh
Hi
which='which $0'
mydir='dirname $which'

. ${mydir}/cs_common.sh

XIL_CS_CLASSPATH="-classpath ${XIL_CS_JAR}/jdom.jar:${XIL_CS_JAR}/jbcl.jar:${XIL_CS_JAR}/chipscope.jar:${XIL_CS_JAR}/ipce.jar"

$XIL_CS_JAVA -Xmx512m -Xss4m ${XIL_CS_JVMARGS} ${XIL_CS_CLASSPATH} com.xilinx.chipscope.inserter.InserterApp -server "$@"
```



```
File Edit View Search Terminal Help
~/bin/bash
which='which $0'
mydir='dirname $which'

. ${mydir}/cs_common.sh

XIL_CS_CLASSPATH="-classpath ${XIL_CS_JAR}/jdom.jar:${XIL_CS_JAR}/jbcl.jar:${XIL_CS_JAR}/chipscope.jar:${XIL_CS_JAR}/ipce.jar"

$XIL_CS_JAVA -Xmx512m -Xss4m ${XIL_CS_JVMARGS} ${XIL_CS_CLASSPATH} com.xilinx.chipscope.inserter.InserterApp -server "$@"
```

해당 윗 부분의 ../sh 를 ../bash 로 변경

이후 source ~/.bashrc