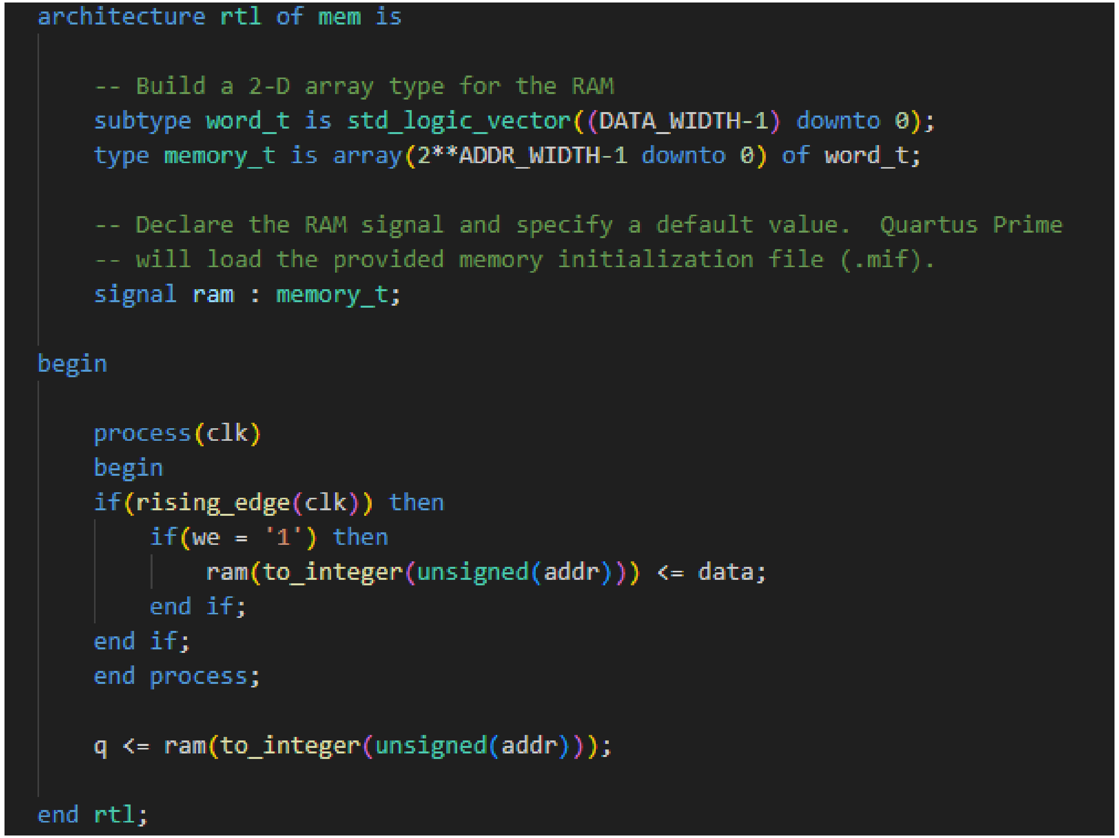
CprE 381: Computer Organization and Assembly Level Programming

Lab Week 4 VHDL

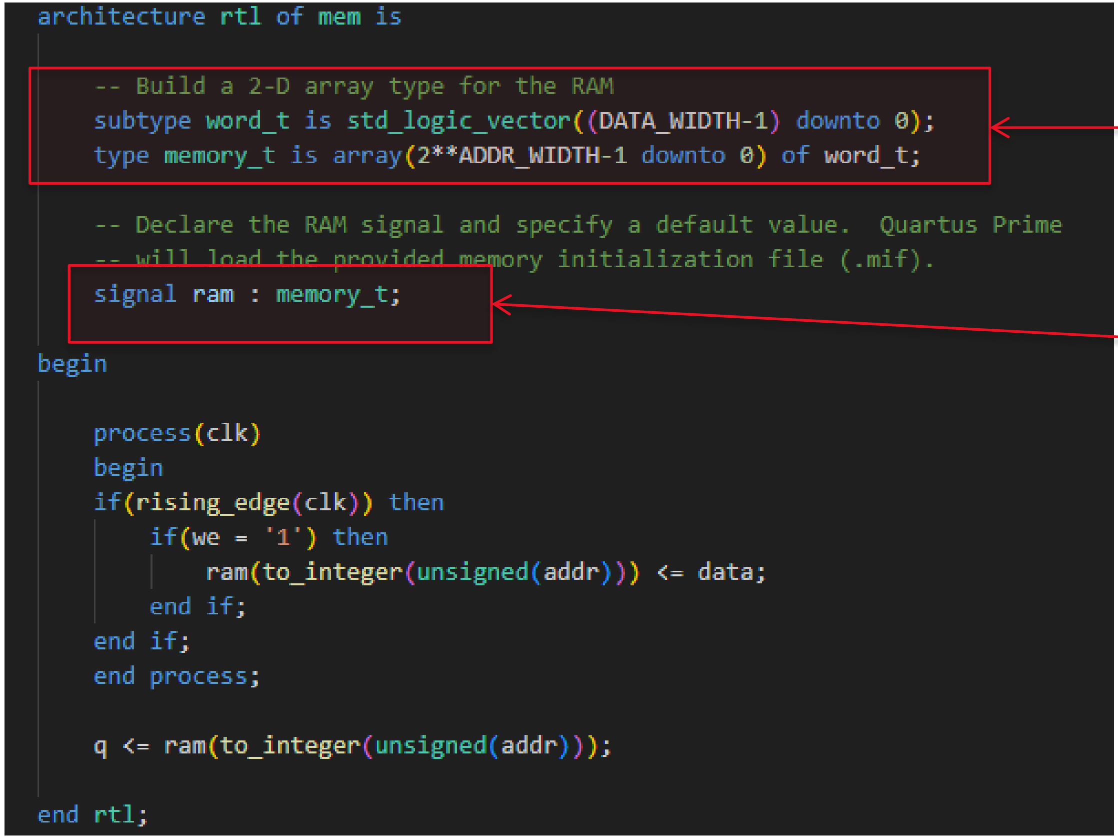
Henry Duwe
Electrical and Computer Engineering
Iowa State University

Modeling Memory in VHDL



```vhdl
architecture rtl of mem is

    -- Build a 2-D array type for the RAM
    subtype word_t is std_logic_vector((DATA_WIDTH-1) downto 0);
    type memory_t is array(2**ADDR_WIDTH-1 downto 0) of word_t;

    -- Declare the RAM signal and specify a default value.  Quartus Prime
    -- will load the provided memory initialization file (.mif).
    signal ram : memory_t;

begin

    process(clk)
    begin
    if(rising_edge(clk)) then
        if(we = '1') then
            ram(to_integer(unsigned(addr))) <= data;
        end if;
    end if;
    end process;

    q <= ram(to_integer(unsigned(addr)));

end rtl;
```

Modeling Memory in VHDL

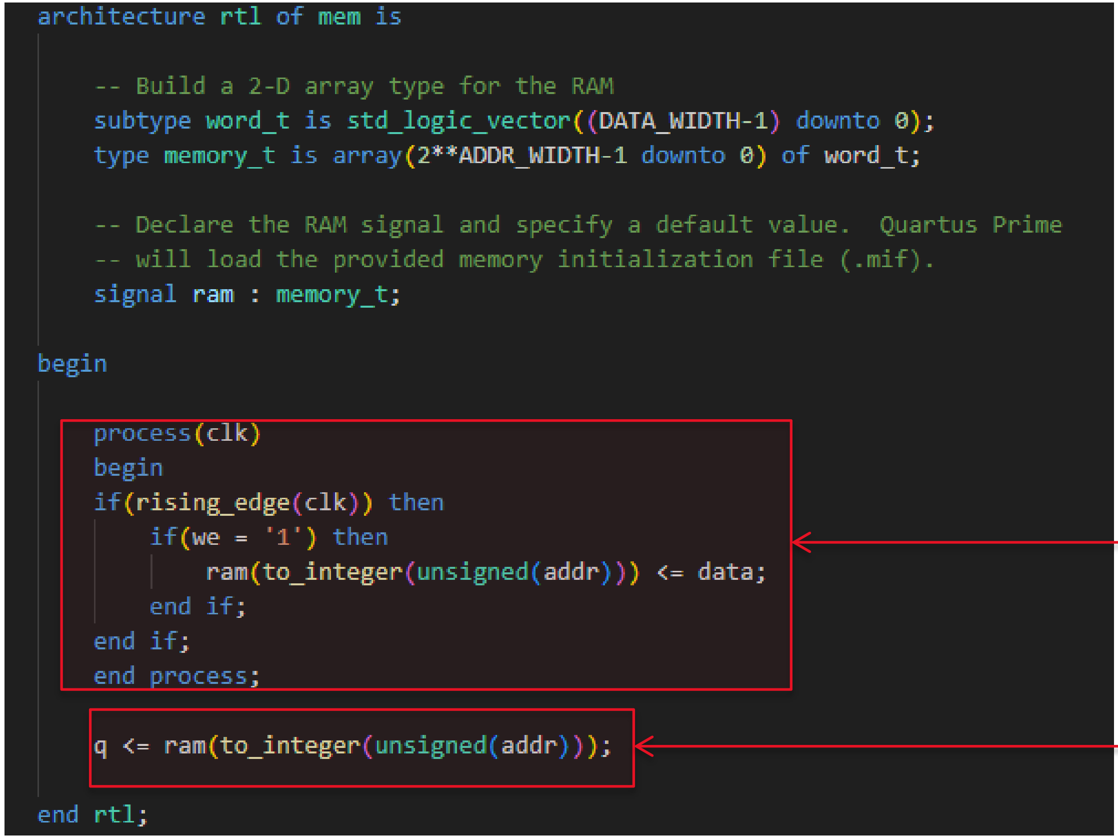


```
architecture rtl of mem is

    -- Build a 2-D array type for the RAM
    subtype word_t is std_logic_vector((DATA_WIDTH-1) downto 0);
    type memory_t is array(2**ADDR_WIDTH-1 downto 0) of word_t;

    -- Declare the RAM signal and specify a default value.  Quartus Prime
    -- will load the provided memory initialization file (.mif).
    signal ram : memory_t;

begin

    process(clk)
    begin
    if(rising_edge(clk)) then
        if(we = '1') then
            ram(to_integer(unsigned(addr))) <= data;
        end if;
    end if;
    end process;

    q <= ram(to_integer(unsigned(addr)));

end rtl;
```

Defining data type

Instantiation

Modeling Memory in VHDL



```vhdl
architecture rtl of mem is

    -- Build a 2-D array type for the RAM
    subtype word_t is std_logic_vector((DATA_WIDTH-1) downto 0);
    type memory_t is array(2**ADDR_WIDTH-1 downto 0) of word_t;

    -- Declare the RAM signal and specify a default value.  Quartus Prime
    -- will load the provided memory initialization file (.mif).
    signal ram : memory_t;

begin

    process(clk)
    begin
    if(rising_edge(clk)) then
        if(we = '1') then
            ram(to_integer(unsigned(addr))) <= data;
        end if;
    end if;
    end process;

    q <= ram(to_integer(unsigned(addr)));

end rtl;
```

Synchronous write

Asynchronous read

Word Vs. Byte Addressable memory

Word Addressable

- This is the format provided in lab
- Must read/write 4 bytes at a time
- Only 4 addresses (0x00-0x03)
-Big Endian format

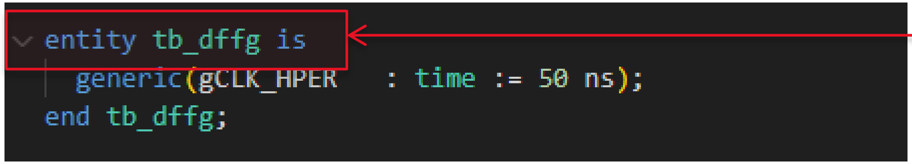
00	0x14120900
01	0x44220215
10	0xFFFFFFFF
11	0x00000000

Byte Addressable

- This is the format RISC-V uses
- Can read/write to individual bits
- 16 addresses
  (0x00-0x0F)

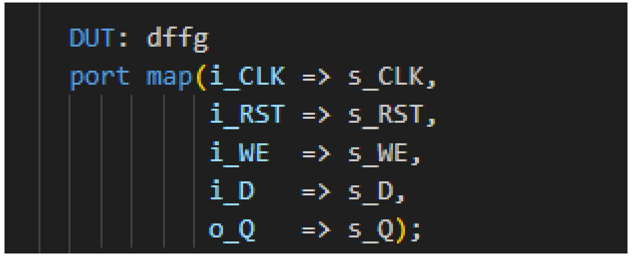
	00	01	10	11
00	0x00	0x09	0x12	0x14
01	0x15	0x02	0x22	0x44
10	0xFF	0xFF	0xFF	0xFF
11	0x00	0x00	0x00	0x00

Simulation Hierarchy

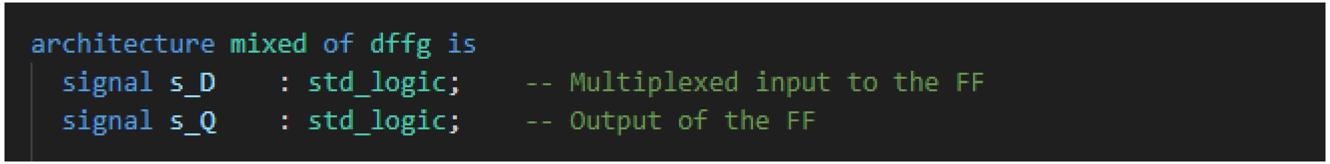


```vhdl
entity tb_dffg is
  generic(gCLK_HPER   : time := 50 ns);
end tb_dffg;
```

Simulated Entity

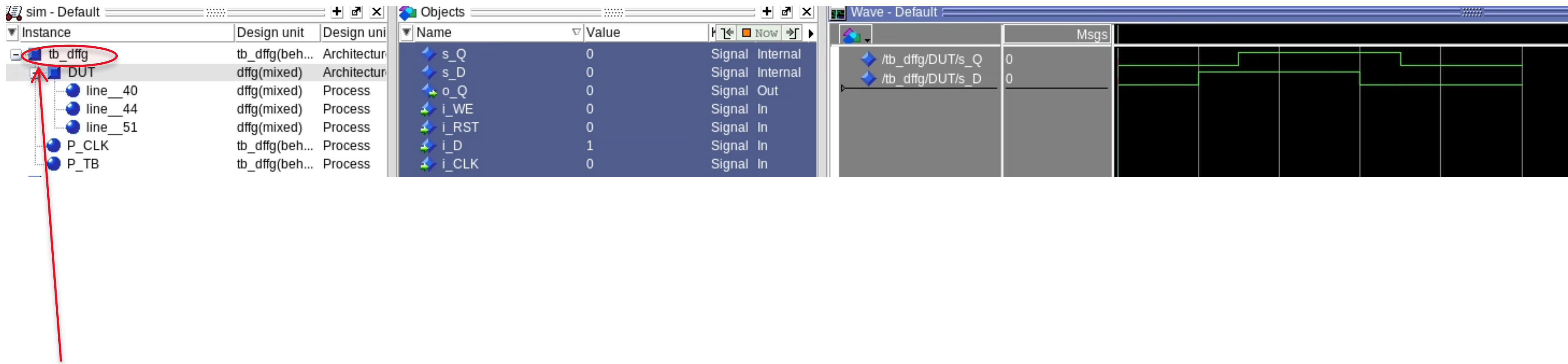


```vhdl
  DUT: dffg
  port map(i_CLK => s_CLK,
           i_RST => s_RST,
           i_WE  => s_WE,
           i_D   => s_D,
           o_Q   => s_Q);
```



```vhdl
architecture mixed of dffg is
  signal s_D    : std_logic;    -- Multiplexed input to the FF
  signal s_Q    : std_logic;    -- Output of the FF
```

Simulation Hierarchy



sim - Default

Instance	Design unit	Design uni
tb_dffg	tb_dffg(beh...	Architectur
DUT	dffg(mixed)	Architectur
line__40	dffg(mixed)	Process
line__44	dffg(mixed)	Process
line__51	dffg(mixed)	Process
P_CLK	tb_dffg(beh...	Process
P_TB	tb_dffg(beh...	Process

Objects

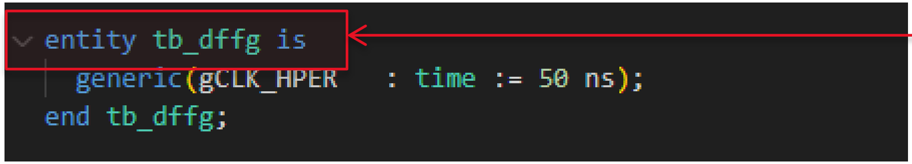
Name	Value		
s_Q	0	Signal	Internal
s_D	0	Signal	Internal
o_Q	0	Signal	Out
i_WE	0	Signal	In
i_RST	0	Signal	In
i_D	1	Signal	In
i_CLK	0	Signal	In

Wave - Default

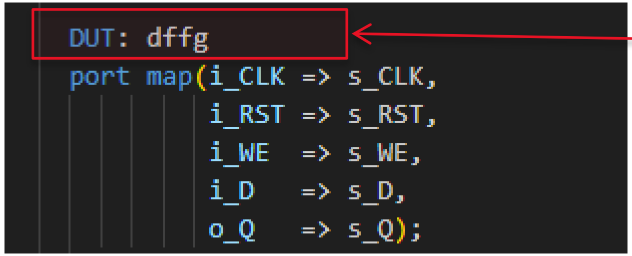
	Msgs
/tb_dffg/DUT/s_Q	0
/tb_dffg/DUT/s_D	0

Simulated Entity

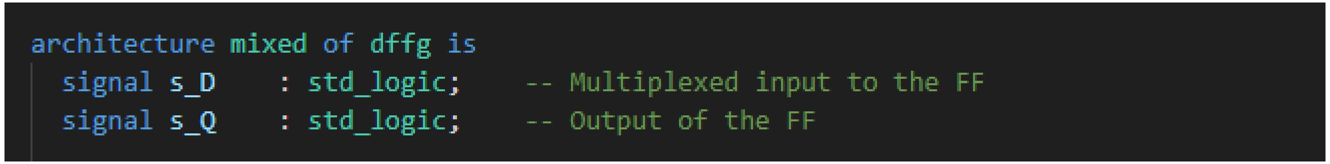
Simulation Hierarchy



```
entity tb_dffg is
  generic(gCLK_HPER    : time := 50 ns);
end tb_dffg;
```

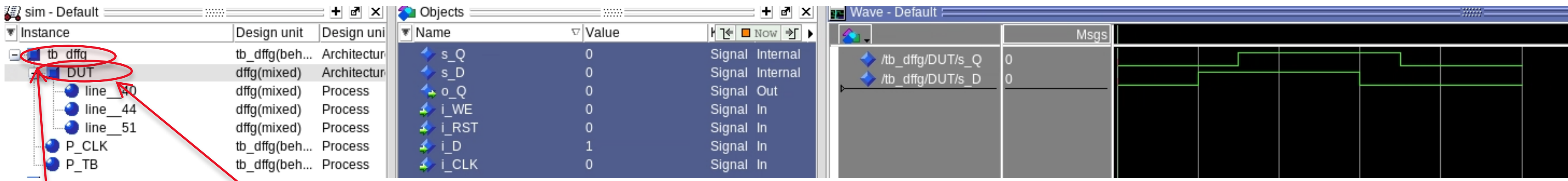


```
  DUT: dffg
  port map(i_CLK => s_CLK,
           i_RST => s_RST,
           i_WE  => s_WE,
           i_D   => s_D,
           o_Q   => s_Q);
```



```
architecture mixed of dffg is
  signal s_D     : std_logic;     -- Multiplexed input to the FF
  signal s_Q     : std_logic;     -- Output of the FF
```

Simulation Hierarchy



Instance	Design unit	Design uni
tb_dffg	tb_dffg(beh...	Architectur
DUT	dffg(mixed)	Architectur
line__40	dffg(mixed)	Process
line__44	dffg(mixed)	Process
line__51	dffg(mixed)	Process
P_CLK	tb_dffg(beh...	Process
P_TB	tb_dffg(beh...	Process

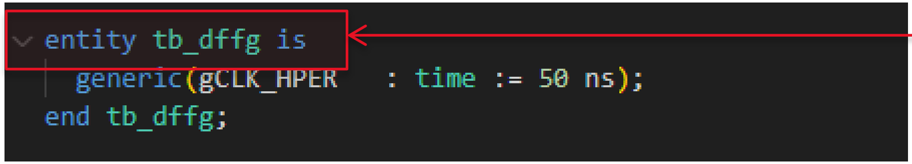
Name	Value		
s_Q	0	Signal	Internal
s_D	0	Signal	Internal
o_Q	0	Signal	Out
i_WE	0	Signal	In
i_RST	0	Signal	In
i_D	1	Signal	In
i_CLK	0	Signal	In

	Msgs
/tb_dffg/DUT/s_Q	0
/tb_dffg/DUT/s_D	0

Component Instantiation

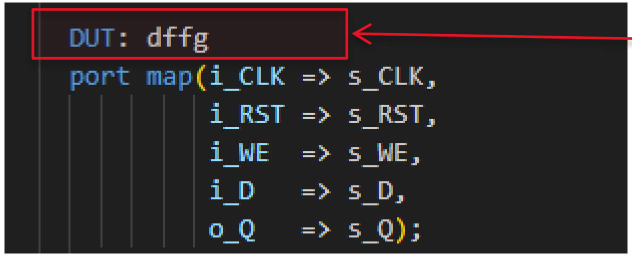
Simulated Entity

Simulation Hierarchy



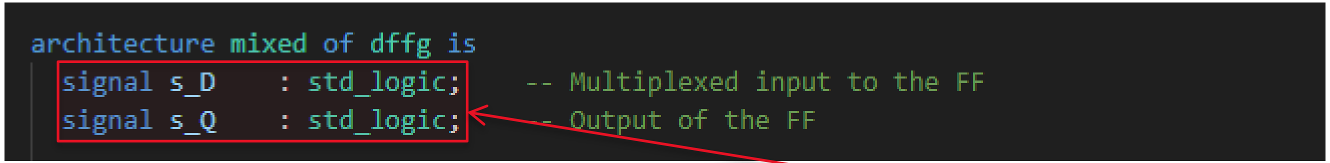
```
entity tb_dffg is
  generic(gCLK_HPER   : time := 50 ns);
end tb_dffg;
```

Simulated Entity



```
  DUT: dffg
  port map(i_CLK => s_CLK,
           i_RST => s_RST,
           i_WE  => s_WE,
           i_D   => s_D,
           o_Q   => s_Q);
```

Component Instantiation

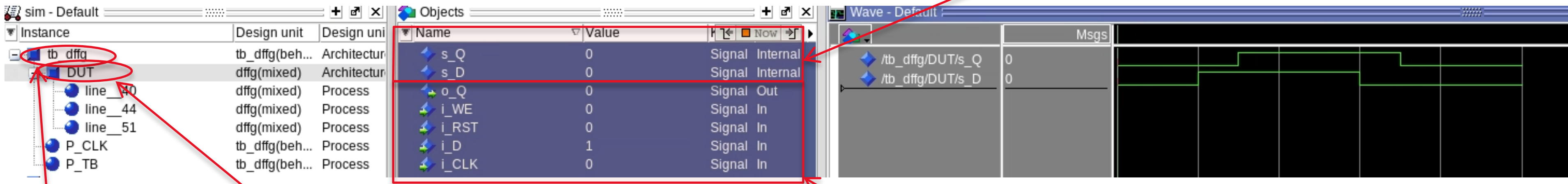


```
architecture mixed of dffg is
  signal s_D     : std_logic;    -- Multiplexed input to the FF
  signal s_Q     : std_logic;    -- Output of the FF
```

Internal Signals

Simulation Hierarchy

Internal Signals



Instance	Design unit	Design uni
tb_dffg	tb_dffg(beh...	Architectur
DUT	dffg(mixed)	Architectur
line__40	dffg(mixed)	Process
line__44	dffg(mixed)	Process
line__51	dffg(mixed)	Process
P_CLK	tb_dffg(beh...	Process
P_TB	tb_dffg(beh...	Process

Name	Value		
s_Q	0	Signal	Internal
s_D	0	Signal	Internal
o_Q	0	Signal	Out
i_WE	0	Signal	In
i_RST	0	Signal	In
i_D	1	Signal	In
i_CLK	0	Signal	In

	Msgs
/tb_dffg/DUT/s_Q	0
/tb_dffg/DUT/s_D	0

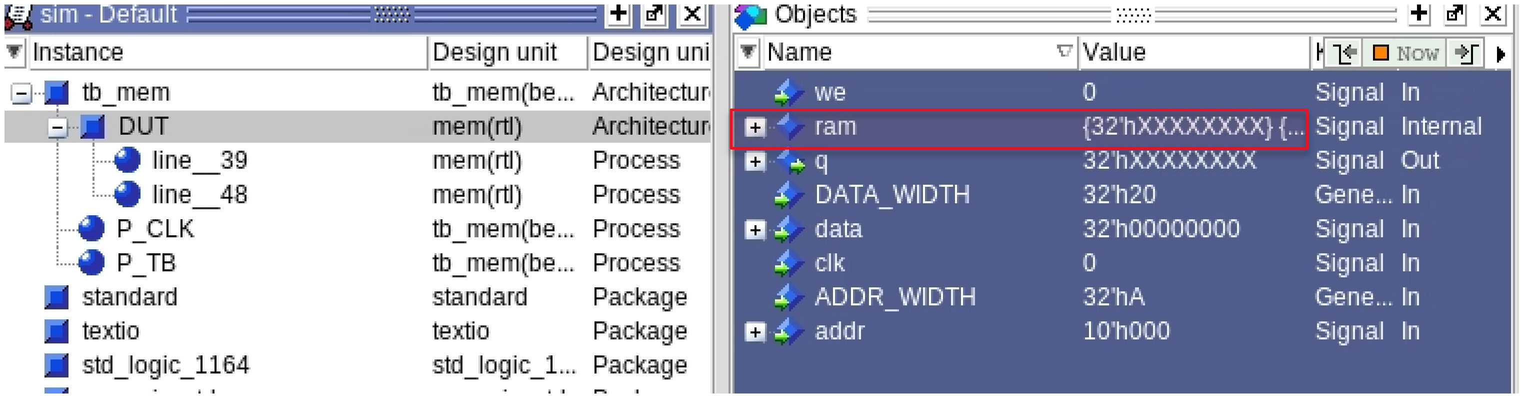
Component Instantiation

Internal Ports

Simulated Entity

- You can find the address of a component by following this hierarchy
- Ex, the address of the internal signal s_Q is “/tb_dffg/DUT/s_Q”

Initializing memory



sim - Default

Instance	Design unit	Design uni
tb_mem	tb_mem(be...	Architectur
DUT	mem(rtl)	Architectur
line__39	mem(rtl)	Process
line__48	mem(rtl)	Process
P_CLK	tb_mem(be...	Process
P_TB	tb_mem(be...	Process
standard	standard	Package
textio	textio	Package
std_logic_1164	std_logic_1...	Package

Objects

Name	Value		
we	0	Signal	In
ram	{32'hXXXXXXXX} {...	Signal	Internal
q	32'hXXXXXXXX	Signal	Out
DATA_WIDTH	32'h20	Gene...	In
data	32'h00000000	Signal	In
clk	0	Signal	In
ADDR_WIDTH	32'hA	Gene...	In
addr	10'h000	Signal	In

mem load –infile {path to memory file} – format hex {path in simulated hierarchy}

- Our goal in this case is to initialize the RAM signal, so we should use the address
  - /tb_mem/DUT/RAM