```
entity FIFO 4x8 is

Port (rst: in STD_LOGIC;

clk: in STD_LOGIC;

addr: in STD_LOGIC_VECTOR (1 downto 0) := "00";

d_in: in STD_LOGIC_VECTOR (7 downto 0);

rd_wr: in STD_LOGIC;

d_out out STD_LOGIC_VECTOR (7 downto 0) := "00000000";

empty: out STD_LOGIC := '1';

full: out STD_LOGIC:= '0');

end FIFO_4x8;

architecture FIFO_4x8_arch of FIFO_4x8 is

TYPE mem IS ARRAY(3 DOWNTO 0) OF STD_LOGIC_VECTOR (7 DOWNTO 0); SIGNAL memory: mem :=
(others=>(others=>'0'));

begin

PROCESS(rst, clk, addr, d_in, rd_wr)

begin
```

```
if rst = '1' then

d_out <= "00000000";

empty <= '1';

full <= '0';

memory <= (others=>(others=>'0'));

elsif falling_edge(clk) then

case rd_wr is

when '0'=>

d_out <= memory(conv_integer(addr)); empty <= '0';

full <=

when others =>

memory(conv_integer(addr)) <=d_in; empty <= '0';

if addr = "11" then
full <= '1';

else end if; full <= '0';

end case, end if;
```

end process;

end FIFO_4x8_arch;

tbbb

-Inputs

signal rst: std_logic := '0'; signal clk: std_logic := '1'; signal addr: std_logic_vector(1 downto 0) := (others => '0'); signal d_in: std_logic_vector(7 downto 0) := (others => '0'); signal rd_wr: std_logic := '0';

--Outputs

signal d_out: std_logic_vector(7 downto 0); signal empty: std_logic;

signal full: std_logic;

Clock period definitions constant clk_period: time := 10 ns;

BEGIN

- Instantiate the Unit Under Test (UUT)

uut: FIFO_4x8 PORT MAP (rst => rst, clk => clk, addr => addr, d_in => d_in,

rd_wr => rd_wr.

d_out => d_out,

empty => empty,

full=>full);

```
Clock process definitions

clk_process process begin

clk <= not(clk);

end process;

wait for clk_period/2;

- Stimulus process

begin

stim_proc: process

rst <= '0';

wait for clk_period;

rst <= '1';

wait for clk_period;

rd_wr <= '1';

for address in 0 to 3 loop

addr <= std_logic_vector(to_unsigned(address , 2));
```

```
d_in <= std_logic_vector(to_unsigned(63*(address + 1), 8)); wait for clk_period*2;

end loop;

d_in <= std_logic_vector(to_unsigned(0, 8));

rd_wr <= '0';

for address in 0 to 3 loop

addr <= std_logic_vector(to_unsigned(address, 2)); wait for clk_period*2;

end loop;

end process;

END;
```