

---

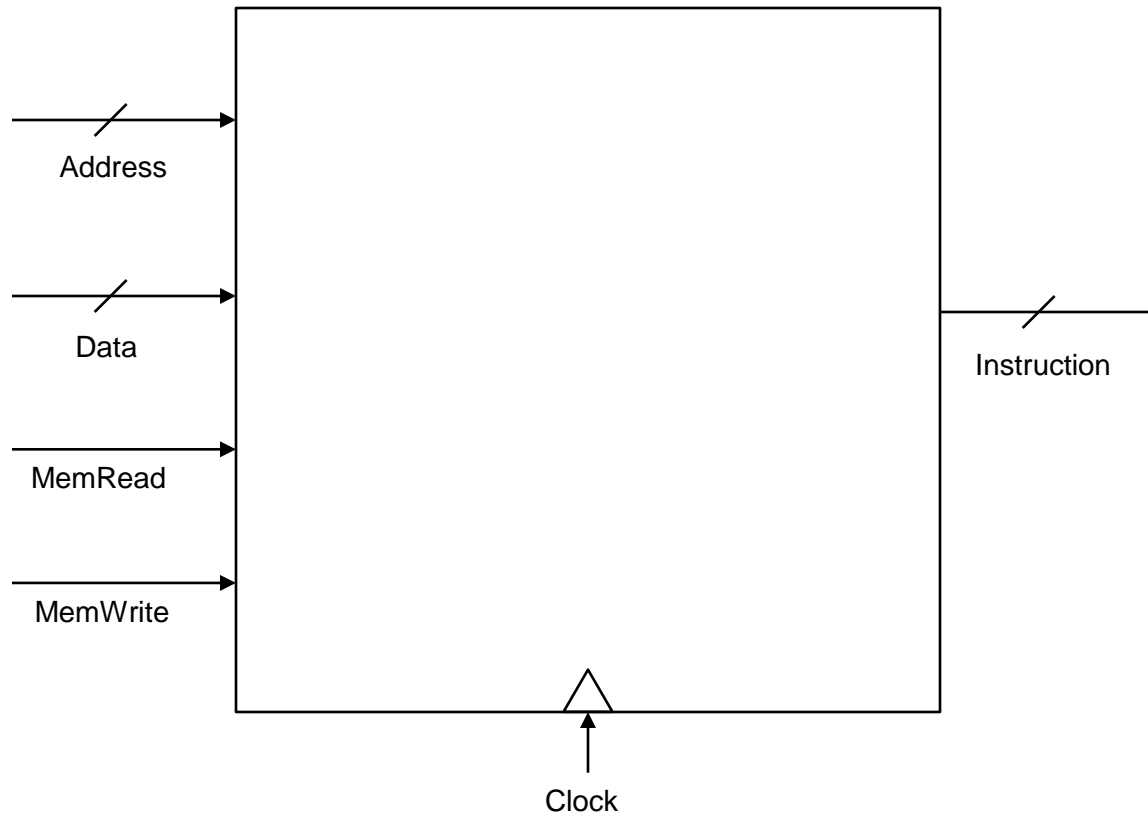
**EE434**  
**ASIC & Digital Systems**

VHDL  
Memory

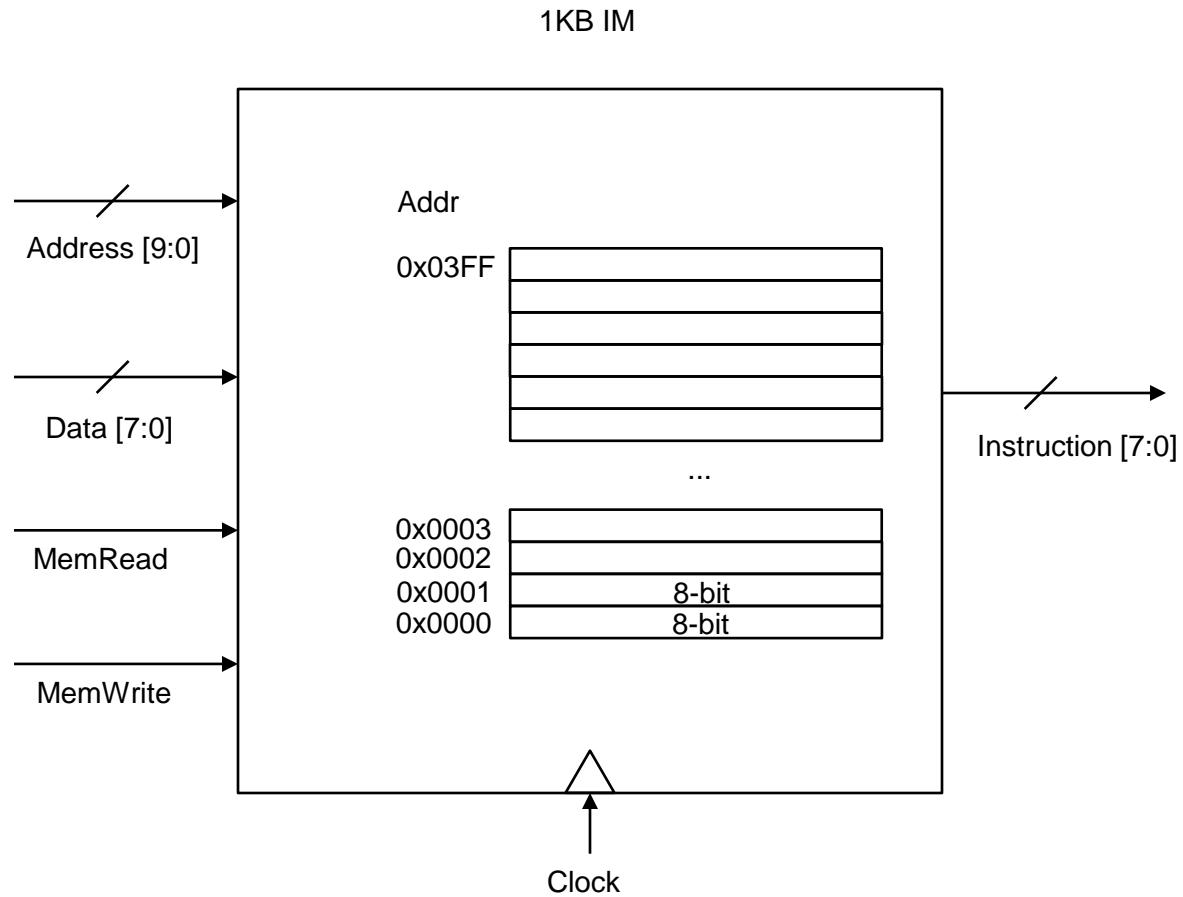
Spring 2016  
Dae Hyun Kim  
daehyun@eecs.wsu.edu

# Instruction Memory

---



# Instruction Memory



# Instruction Memory

---

- Entity name
  - myIM
- Inputs
  - Addr [9:0], Data [7:0], MemRead, MemWrite, Clk
- Output
  - Instr [7:0]

# Instruction Memory

---

- Function (when Clk ↑)
  - If MemRead = 1
    - Instr = Mem[Addr]
  - If MemRead = 0 and MemWrite = 1
    - Mem[Addr] = Data
  - If MemRead = 0 and MemWrite = 0
    - Nothing happens

# Instruction Memory

---

- Implementation

```
ENTITY myIM IS
  PORT ( Addr : IN std_logic_vector (9 DOWNT0 0);
        Data : IN std_logic_vector (7 DOWNT0 0);
        MemRead, MemWrite, Clk : IN std_logic;
        Instr : OUT std_logic_vector (7 DOWNT0 0));
END myIM;
```

# Instruction Memory

- Implementation

```
ARCHITECTURE myIM_arch OF myIM IS
  SUBTYPE byte IS std_logic_vector (7 DOWNT0 0);
  TYPE t_1KB IS ARRAY (0 TO 1023) OF byte;
  SIGNAL mem : t_1KB;
BEGIN
  PROCESS (Clk)
    VARIABLE ind : unsigned (9 DOWNT0 0);
  BEGIN
    IF rising_edge (Clk) THEN
      ind := unsigned (Addr);
      IF (MemRead = '1') THEN
        Instr <= mem (to_integer(ind));
      ELSIF (MemWrite = '1') THEN
        mem(to_integer(ind)) <= Data;
      END IF;
    END IF;
  END PROCESS;
END myIM_arch;
```

Use "SUBTYPE" if you define an existing type.

Include "USE IEEE.numeric\_std.all".

- Test

```
u1 : myOSC PORT MAP (n_clk);
u2 : myIM PORT MAP (n_addr, n_data, n_mem_read, n_mem_write, n_clk, n_instr);
PROCESS
BEGIN

  n_addr <= "0000000000";
  n_data <= "00000000";
  n_mem_read <= '0';
  n_mem_write <= '0';
  WAIT FOR 0.7 NS;
  n_mem_write <= '1';
  n_data <= "01010101";
  WAIT FOR 1 NS;
  n_addr <= "0000000001";
  n_data <= "10101010";
  WAIT FOR 1 NS;
  n_mem_read <= '1';
  n_mem_write <= '0';
  n_addr <= "0000000000";
  WAIT FOR 1 NS;
  n_addr <= "0000000010";
  WAIT FOR 1 NS;
  n_addr <= "0000000001";
  WAIT FOR 1 NS;
END PROCESS;
```



# Instruction Memory

