Digital Systems I

Lab08 Slides based on **Chapter 7 - Part II**

Behavioral Modeling

Process Constructs

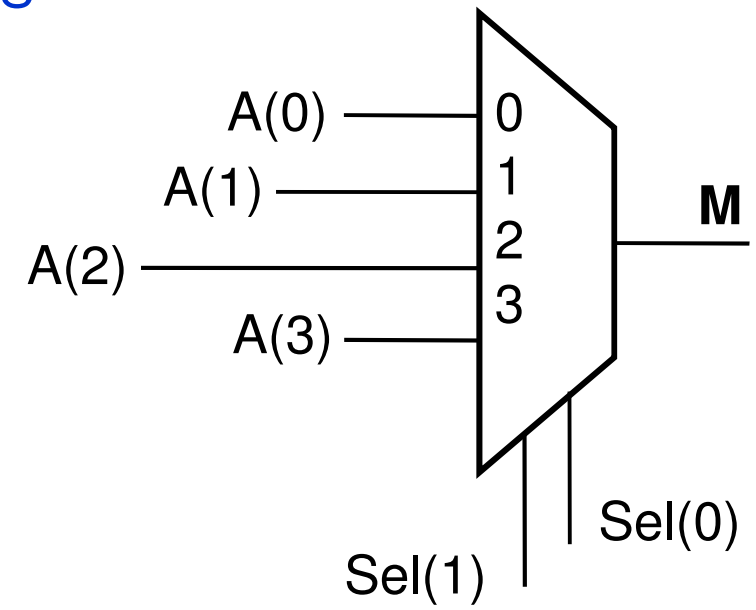
IF THEN ELSE Statement

N. Tabrizi

Kettering University

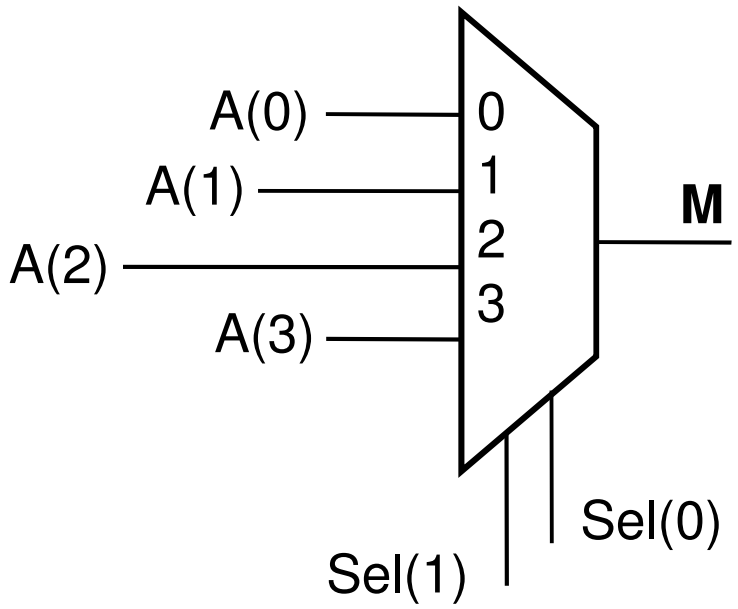
IF-THEN-ELSE Statement

```
ENTITY mux4_1 IS
    PORT (A     : IN     STD_LOGIC_VECTOR (3 DOWNTO 0);
          Sel   : IN     STD_LOGIC_VECTOR (1 DOWNTO 0);
          M     : OUT    STD_LOGIC
         );
END mux4_1;

ARCHITECTURE Behavior OF mux4_1 IS
BEGIN
.
.
.
END Behavior;
```



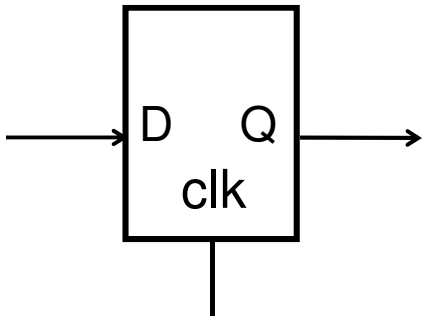
IF-THEN-ELSE Statement

```
ARCHITECTURE Behavior OF mux4_1 IS
BEGIN
    PROCESS (A, Sel)              -- *** Sensitivity list ***
    BEGIN
        IF      Sel = "00"        THEN   M <= A(0);
        ELSIF   Sel = "01"        THEN
                M <= A(1);
        ELSIF   Sel = "10"        THEN
                M <= A(2);
        ELSE
                M <= A(3);
        END IF;
    END PROCESS;
END Behavior;
```



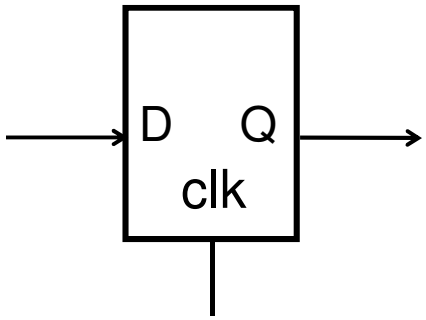
D-latch

```
ENTITY dmem IS
    PORT ( D, clk : IN      STD_LOGIC;
                 Q   : OUT    STD_LOGIC
           );
END dmem;

ARCHITECTURE Behavior OF dmem IS
BEGIN
    PROCESS (D, clk)
    BEGIN
        IF clk = '1' THEN
            Q <= D;
        ELSE
            Q <= Q;
        END IF;

    END PROCESS;
END Behavior;
```

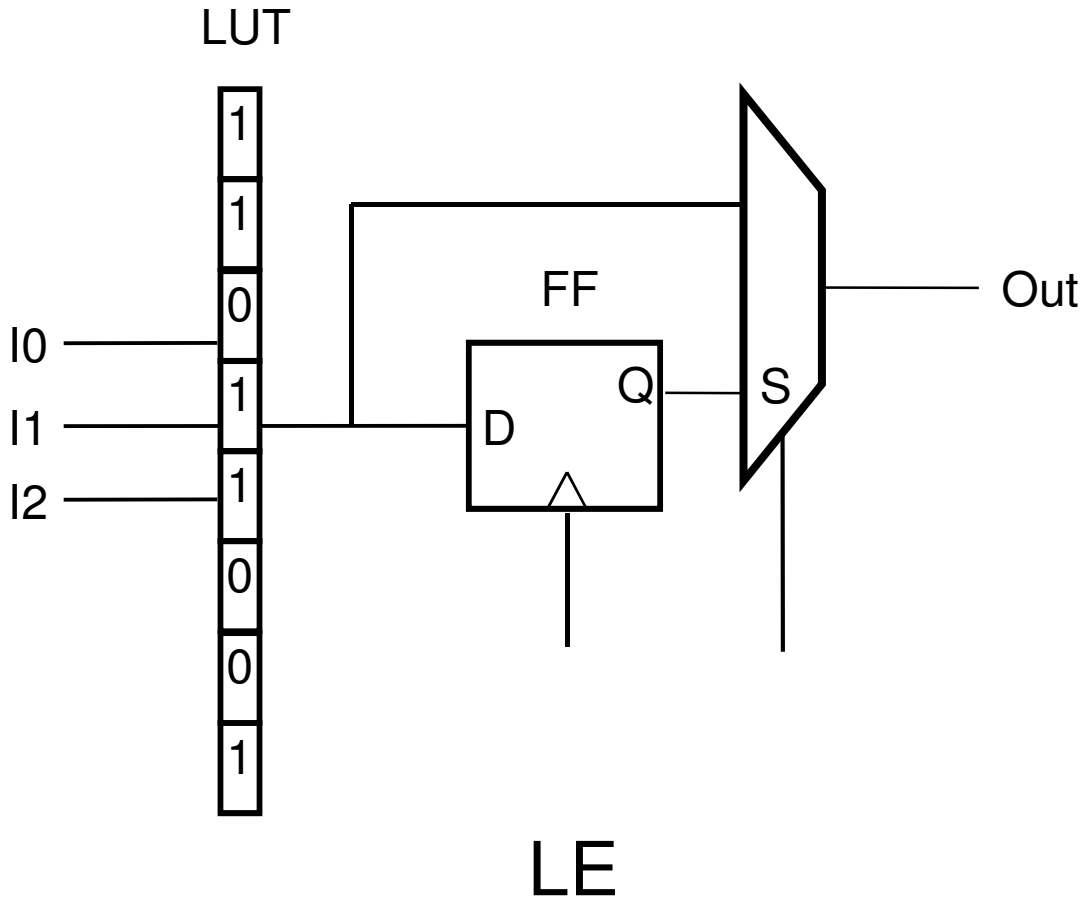


D-latch

```
ENTITY dmem IS
    PORT ( D, clk : IN      STD_LOGIC;
                 Q   : OUT    STD_LOGIC
           );
END dmem;

ARCHITECTURE Behavior OF dmem IS
BEGIN
    PROCESS (D, clk)
    BEGIN
        IF clk = '1' THEN
            Q <= D;


        END IF;
    END PROCESS;
END Behavior;
```

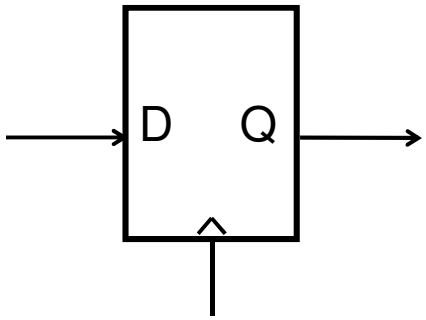


Do not build FFs!



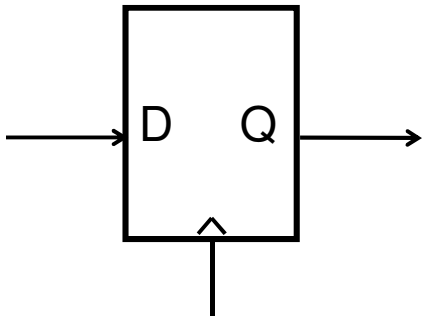
D-FF

```
ENTITY dmem IS
    PORT ( D, clk : IN      STD_LOGIC;
                 Q   : OUT    STD_LOGIC
           );
END dmem;

ARCHITECTURE Behavior OF dmem IS
BEGIN
    PROCESS (clk)
    BEGIN
        IF clk'EVENT AND clk = '1' THEN
                Q <= D;
        ELSE
                Q <= Q;
        END IF;

    END PROCESS;
END Behavior;
```



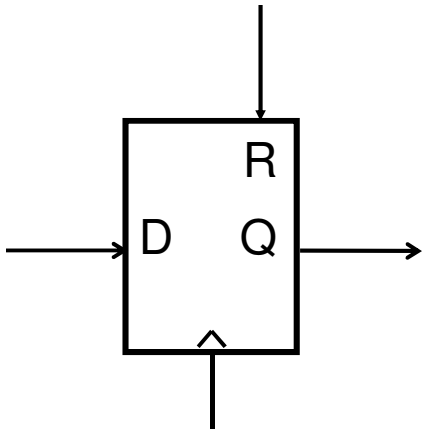
D-FF

```
ENTITY dmem IS
    PORT ( D, clk : IN      STD_LOGIC;
                 Q   : OUT    STD_LOGIC
           );
END dmem;

ARCHITECTURE Behavior OF dmem IS
BEGIN
    PROCESS (clk)
    BEGIN
        IF clk'EVENT AND clk = '1' THEN
                Q <= D;


        END IF;

    END PROCESS;
END Behavior;
```



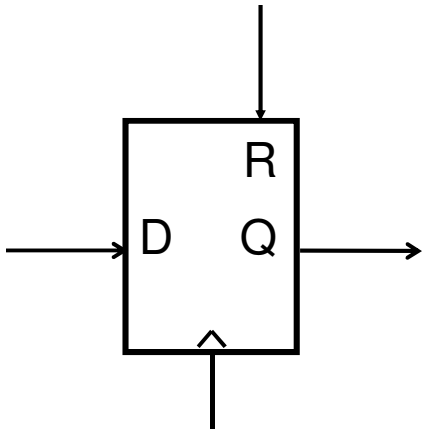
D-FF with asynchronous reset

```
ARCHITECTURE Behavior OF dmem IS
BEGIN
    PROCESS (clk, R)  -- R is asynchronous
    BEGIN
        IF R = '1' THEN
                Q <= '0';
        ELSIF clk'EVENT AND clk = '1' THEN
                Q <= D;
        ELSE
                Q <= Q;
        END IF;
    END PROCESS;
END Behavior;
```



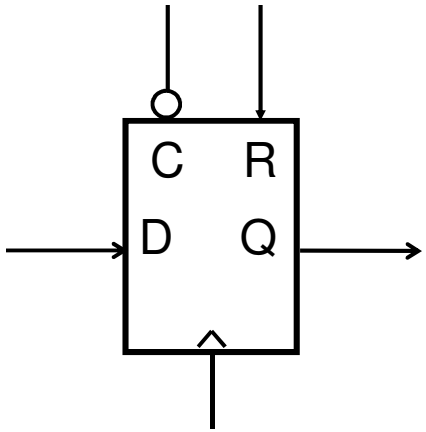
D-FF with asynchronous reset

```
ARCHITECTURE Behavior OF dmem IS
BEGIN
    PROCESS (clk, R)  -- R is asynchronous
    BEGIN
        IF R = '1' THEN
                Q <= '0';
        ELSIF clk'EVENT AND clk = '1' THEN
                Q <= D;


        END IF;
    END PROCESS;
END Behavior;
```



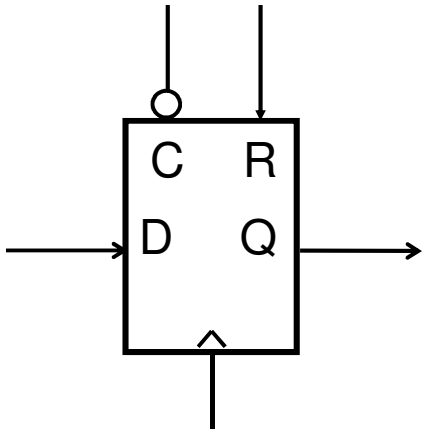
D-FF with asynchronous and synchronous resets

```
ARCHITECTURE Behavior OF dmem IS
BEGIN
    PROCESS (clk, R)  -- R is asynchronous
    BEGIN
        IF R = '1' THEN
                Q <= '0';
        ELSIF  clk'EVENT AND clk = '1'  THEN
                IF C_n = '0' THEN
                        Q <= '0';
                ELSE
                        Q <= D;
                END IF;
        ELSE
                Q <= Q;
        END IF;
    END PROCESS;
END Behavior;
```



D-FF with asynchronous and synchronous resets

```
ARCHITECTURE Behavior OF dmem IS
BEGIN
    PROCESS (clk, R)  -- R is asynchronous
    BEGIN
        IF R = '1' THEN
                Q <= '0';
        ELSIF  clk'EVENT AND clk = '1'  THEN
                IF C_n = '0' THEN
                        Q <= '0';
                ELSE
                        Q <= D;
                END IF;


        END IF;
    END PROCESS;
END Behavior;
```



End of Chapter 7, Part II