Sequential Design

ELEC 311

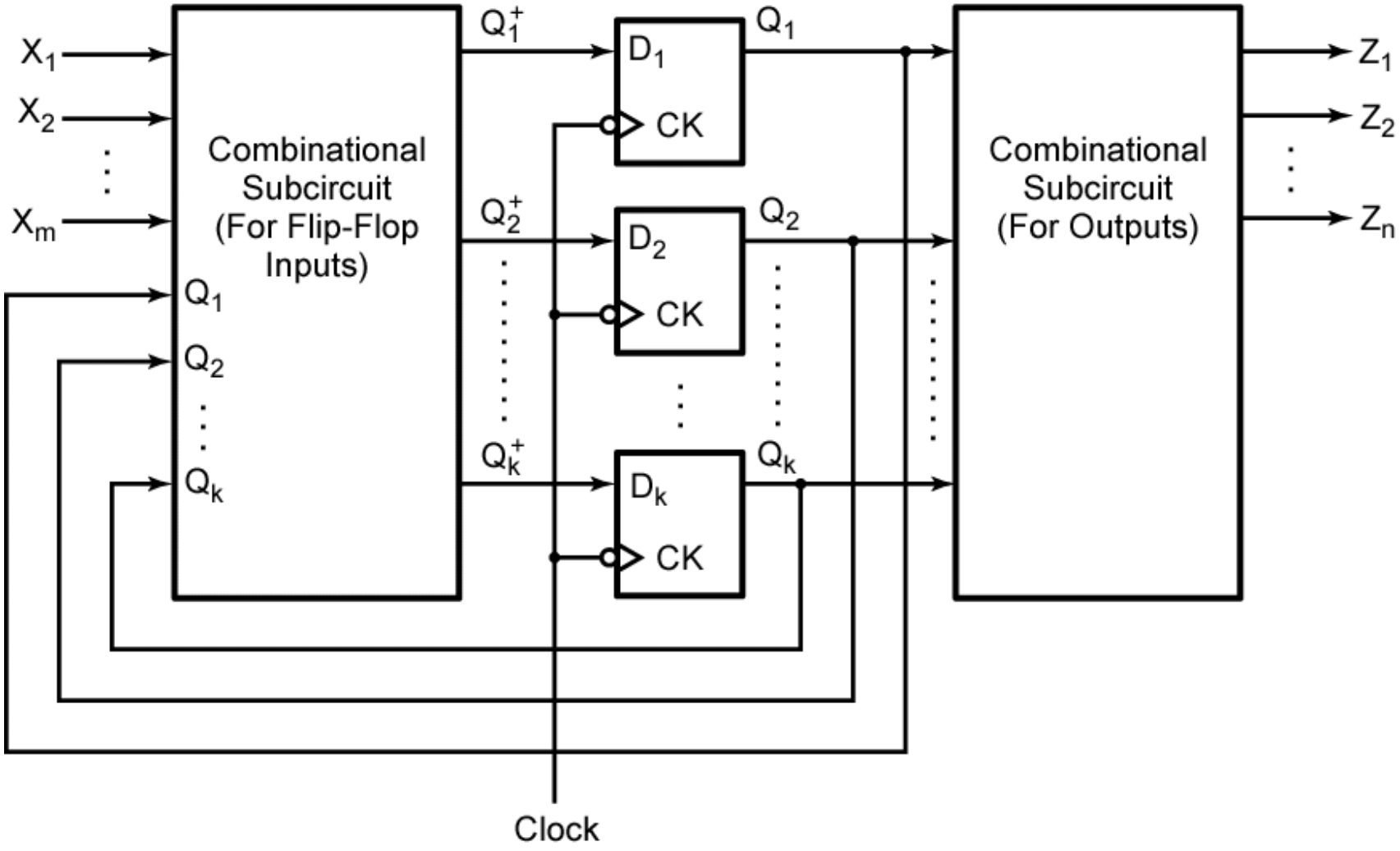
Digital Logic and Circuits

Dr. Ron Hayne

Images Courtesy of Cengage Learning

THE CITADEL
DEPARTMENT OF ELECTRICAL & COMPUTER ENGINEERING

Sequential Circuits



Design Example

- 2-bit Variable Counter

X1	X0	Function	Ex. Sequence (Z1 Z0)
0	0	down	00, 11, 10, 01, 00, …
0	1	up	00, 01, 10, 11, 00, …
1	0	down, even	00, 10, 00, 10, 00, …
1	1	up, odd	01, 11, 01, 11, 01, …

State Assignment: Q1 Q0 = Z1 Z0

Transition Table (1)

	Inputs		PS		NS	
mt	X1	X0	Q1	Q0	Q1+	Q0+
0	0	0	0	0		
1	0	0	0	1		
2	0	0	1	0		
3	0	0	1	1		
4	0	1	0	0		
5	0	1	0	1		
6	0	1	1	0		
7	0	1	1	1		

Transition Table (1)

	Inputs		PS		NS	
mt	X1	X0	Q1	Q0	Q1+	Q0+
0	0	0	0	0	1	1
1	0	0	0	1	0	0
2	0	0	1	0	0	1
3	0	0	1	1	1	0
4	0	1	0	0		
5	0	1	0	1		
6	0	1	1	0		
7	0	1	1	1		

Transition Table (1)

	Inputs		PS		NS	
mt	X1	X0	Q1	Q0	Q1+	Q0+
0	0	0	0	0	1	1
1	0	0	0	1	0	0
2	0	0	1	0	0	1
3	0	0	1	1	1	0
4	0	1	0	0	0	1
5	0	1	0	1	1	0
6	0	1	1	0	1	1
7	0	1	1	1	0	0

Transition Table (2)

	Inputs		PS		NS	
mt	X1	X0	Q1	Q0	Q1+	Q0+
8	1	0	0	0		
9	1	0	0	1		
10	1	0	1	0		
11	1	0	1	1		
12	1	1	0	0		
13	1	1	0	1		
14	1	1	1	0		
15	1	1	1	1		

Transition Table (2)

	Inputs		PS		NS	
mt	X1	X0	Q1	Q0	Q1+	Q0+
8	1	0	0	0	1	0
9	1	0	0	1	0	0
10	1	0	1	0	0	0
11	1	0	1	1	1	0
12	1	1	0	0		
13	1	1	0	1		
14	1	1	1	0		
15	1	1	1	1		

Transition Table (2)

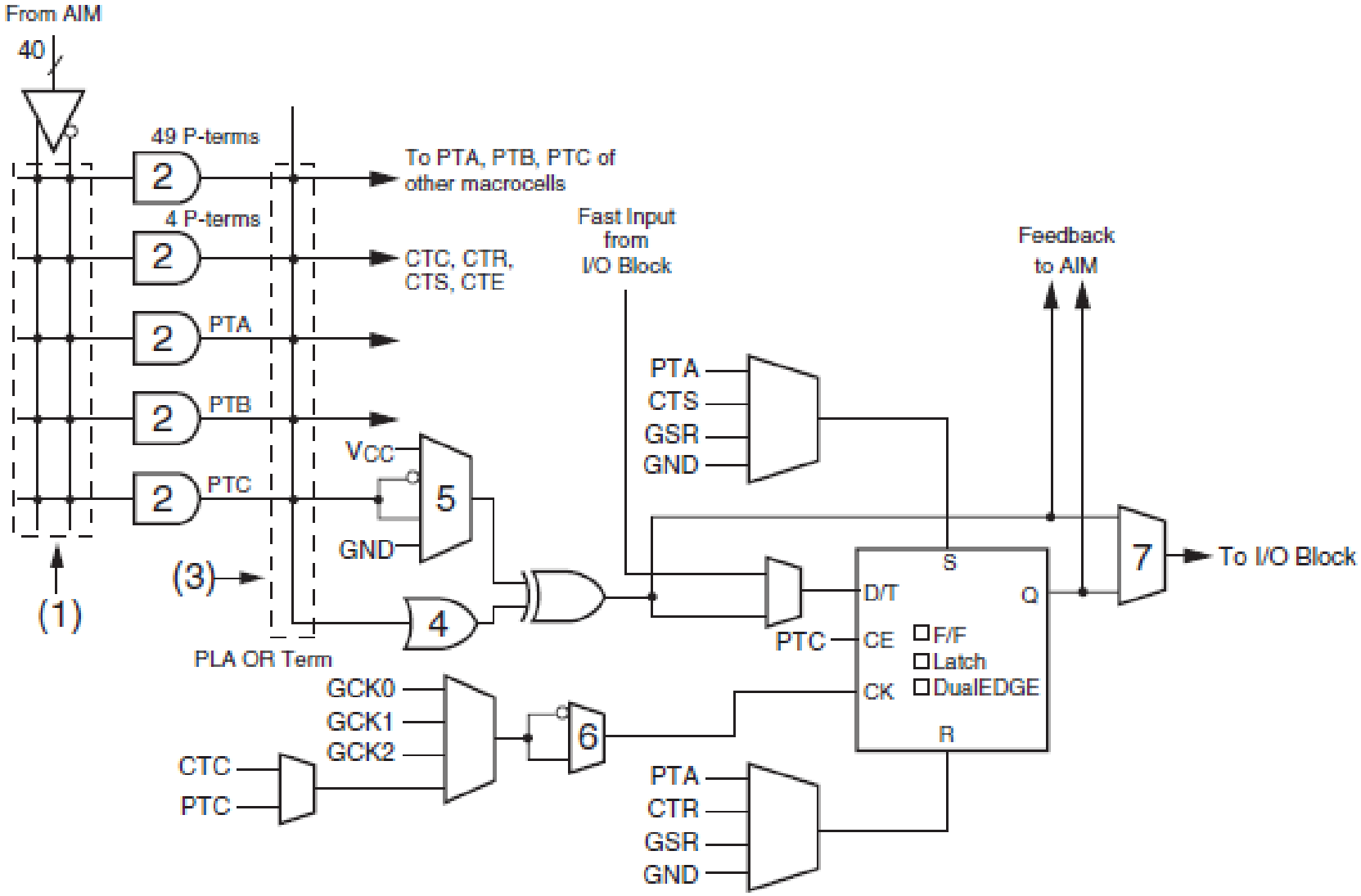
	Inputs		PS		NS	
mt	X1	X0	Q1	Q0	Q1+	Q0+
8	1	0	0	0	1	0
9	1	0	0	1	0	0
10	1	0	1	0	0	0
11	1	0	1	1	1	0
12	1	1	0	0	0	1
13	1	1	0	1	1	1
14	1	1	1	0	1	1
15	1	1	1	1	0	1

Next State (FF) Equations

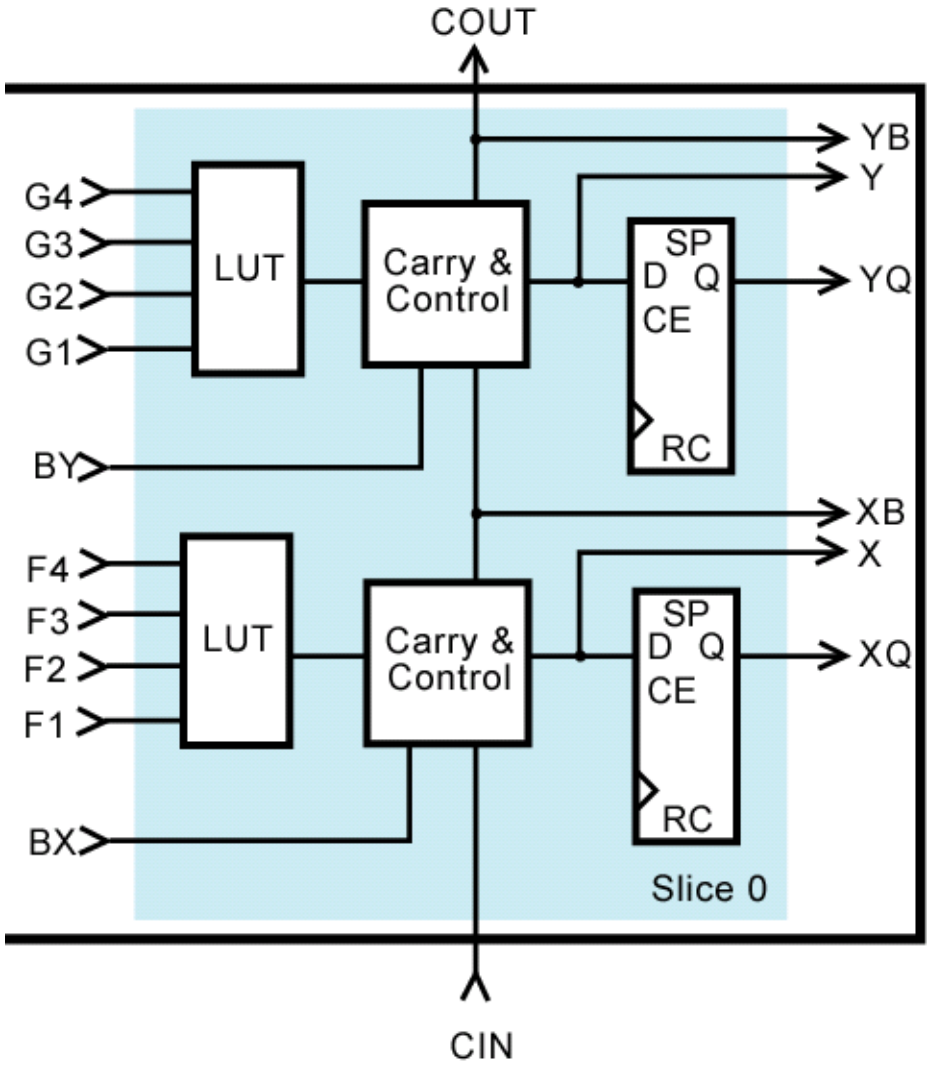
- D1(X1,X0,Q1,Q0) = Q1+ = Σm(0,3,5,6,8,11,13,14)
- D0(X1,X0,Q1,Q0) = Q0+ = Σm(0,2,4,6,12,13,14,15)

- D1 = (X0′·Q1′·Q0′) + (X0·Q1′·Q0) +
  (X0′·Q1·Q0) + (X0·Q1·Q0′)
  - D0 = (X1′·Q0′) + (X1·X0)

Xilinx CPLD Implementation



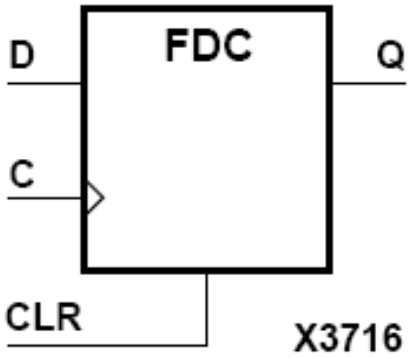
Xilinx FPGA Implementation



UNISIM Library Components

- FDC
  - D Flip-Flop with Asynchronous Clear

```
component FDC
  port (Q : out STD_LOGIC;
        C : in STD_LOGIC;
        CLR : in STD_LOGIC;
        D : in STD_LOGIC);
end component;
```



VHDL Package

- project4_pkg.vhd
  - Clock Divider

```
component CLK_DIV
  port(CLK_IN  : in std_logic;
       CLK_OUT : out std_logic );
end component;
```

VHDL Model

```
library UNISIM;
use UNISIM.VComponents.all;
use work.Project4_Pkg.all;

entity Counter is
  Port ( X : in std_logic_vector(1 downto 0);
         CLR : in std_logic;
         CLK : in std_logic;
         Z : out std_logic_vector(1 downto 0));
end Counter;

architecture DATAFLOW of Counter is
  signal D : std_logic_vector(1 downto 0);
  signal Q : std_logic_vector(1 downto 0);
  signal SLOW_CLK : std_logic;
```

VHDL Model

```
begin

  D(1) <= (not X(0) and not Q(1) and not Q(0)) or
          (X(0) and not Q(1) and Q(0)) or
          (not X(0) and Q(1) and Q(0)) or
          (X(0) and Q(1) and not Q(0));
  D(0) <= (not X(1) and not Q(0)) or
          (X(1) and X(0));
  CDIV : CLK_DIV port map (CLK, SLOW_CLK);
  FF1 : FDC port map(Q(1), SLOW_CLK, CLR, D(1));
  FF0 : FDC port map(Q(0), SLOW_CLK, CLR, D(0));
  Z <= Q;

end DATAFLOW;
```

Summary

- Design Example
  - Functional Description
  - Transition Table
  - Next State Equations
  - FPGA Implementation
  - VHDL Model