

CPE 426/526 Spring 2003
Final Exam Review
with Problem Statements
and Solutions

Dr. Rhonda Kay Gaede

UAH

UAH

CPE 426/526

VHDL Problem Statement

Modify the following VHDL model by adding a parameter that sets the number of flip-flops in the counter. Also, add an input which is loaded with an asynchronous load input signal which is active low.

VHDL

```

use ieee.std_logic_unsigned.all;
entity UPCOUNT is
  port ( CLOCK, RESETN, E : in std_logic;
        Q           : out std_logic_vector (3 downto 0));
end UPCOUNT;
architecture BEHAVIOR of UPCOUNT is
  signal COUNT : std_logic_vector (3 downto 0);
begin
  process (CLOCK, RESETN)
  begin
    if RESETN = '0' then
      COUNT <= '0000';
    elsif (CLOCK'event and CLOCK = '1') then
      if E = '1' then
        COUNT <= COUNT + 1;
      else
        COUNT <= COUNT;
      end if;
    end process; Q <= COUNT;
  end BEHAVIOR;

```

3

VHDL Problem Solution

```

use ieee.std_logic_unsigned.all;

entity UPCOUNT is
  generic (N : integer);
  port ( CLOCK, RESETN, E : in std_logic;
        LD           : in std_logic;
        LD_INPUT     : in std_logic_vector(N-1 downto 0);
        Q           : out std_logic_vector (N-1 downto 0));
end UPCOUNT;

```

4

VHDL Problem Solution (Continued)

```

architecture BEHAVIOR of UPCOUNT is
  signal COUNT : std_logic_vector (N-1 downto 0);
begin
  process (CLOCK, RESETN, LD)
  begin
    if RESETN = '0' then
      COUNT <= (OTHERS => '0');
    elsif (LD = '0') then
      COUNT <= LD_INPUT;
    elsif (CLOCK'event and CLOCK = '1') then
      if E = '1' then
        COUNT <= COUNT + 1;
      else
        COUNT <= COUNT;
      end if;
    end process;
    Q <= COUNT;
  end BEHAVIOR;

```

Synthesis Style

Question:

What kind of hardware element will be inferred by a synthesis tool from the following model?

Answer:

A latch, since the behavior is level sensitive.

Synthesis Style Model

```

entity WIDGET is
  Port (A, B : in SIGNED (0 to 2);
        CLK, RESET : in std_logic;
        Z : out SIGNED(0 to 2));
end WIDGET;
architecture EXAMPLE of WIDGET is
begin
  process (CLK, RESET)
  begin
    if (RESET = '0' then
      Z <= '0';
    elsif (CLK = '0') then
      Z <= A nor B;
    end if;
  end process;
end EXAMPLE;

```

More Synthesis Style

Question:

What kind of hardware element will be inferred by a synthesis tool from the following model?

Answer:

Synopsys: A latch, since $\text{\$EVENT}$ is not present.

Leonardo: A flip-flop, since $\text{\$EVENT}$ is implied.

Second Synthesis Style Model

```

entity WIDGET is
  Port (A, B : in SIGNED (0 to 2);
        CLK : in std_logic;
        Z : out SIGNED(0 to 2));
end WIDGET;
architecture EXAMPLE of WIDGET is
begin
  process (CLK)
  begin
    if (CLK = '1') then
      Z <= A nor B;
    end if;
  end process;
end EXAMPLE;

```

Scheduling Example

Question:

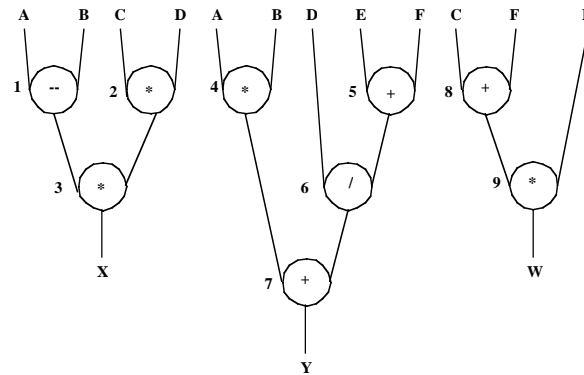
Consider the following VHDL code. Assuming that there are no hardware constraints and that you have an ALU module that performs addition, subtraction, multiplication, and division, draw a data flow graph.

```

entity SCHED2 is
  port (A, B, C, D, E, F: in INTEGER;
        CLK : in BIT;
        W, X, Y: out INTEGER);
end SCHED2;
architecture HIGH_LEVEL of SCHED2 is
  signal Z: INTEGER;
begin
  X <= (A - B) * C * D;
  Y <= (A * B) + (E + F)/D;
  W <= (C + F) * B
end HIGH_LEVEL;

```

Scheduling Example Data Flow Graph



ALAP Scheduling Example

Question:

Consider the following VHDL code. Assuming that you have an two ALU modules that perform addition, subtraction, multiplication, and division, derive an ALAP schedule.

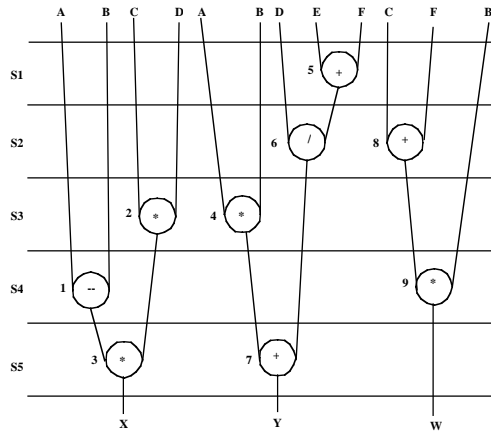
```
entity SCHED2 is
  port (A, B, C, D, E, F: in INTEGER;
        CLK : in BIT;
        W, X, Y: out INTEGER);
end SCHED2;

architecture HIGH_LEVEL of SCHED2 is
  signal Z: INTEGER;
begin
  X <= (A - B) * C * D;
  Y <= (A * B) + (E + F)/D;
  W <= (C + F) * B
end HIGH_LEVEL;
```

ALAP Scheduling Example Answer

Step	Ready List	Scheduled Items
n	{3, 7, 9}	3, 7
n-1	{9, 1, 2, 4, 6, 8}	9, 1
n-2	{2, 4, 6, 8}	2, 4
n-3	{6, 8}	6, 8
n-4	{5}	5

So, n = 5.



Left Edge Algorithm

Question:
For the data lifetime chart shown, use the left edge algorithm to obtain an efficient register allocation.

Answer:

	A	B	C	D	E	F	G	H	I	J	K	L
S1			X	X			X					
S2	X	X			X		X			X		
S3		X				X	X	X		X		
S4	X	X				X			X			X
S5	X					X			X	X	X	X

	A	B	C	D	E	F	G	H	I	J	K	L
S1			X	X			X					
S2	X	X			X		X			X		
S3		X			X	X	X		X			
S4	X	X				X			X			X
S5	X					X			X	X	X	X

	D	C	G	E	J	B	H	F	A	I	L	K
S1	X	X	X									
S2	X	X	X	X	X							
S3		X		X	X	X						
S4					X		X	X	X	X		
S5						X	X	X	X	X		

	R1	R2	R3	R4	R5
S1	D	C	G	-	-
S2	B	C	G	E	J
S3	B	H	G	F	J
S4	B	I	L	F	A
S5	K	I	L	F	A

VHDL Modeling

- Design a resetting sequential majority function that asserts (active high) the output if the past three inputs contain two or more 1's. Assume a Moore machine.

Sample input/output sequences are given below.

- X = 0101110110
- Z = 0000010010

VHDL Modeling Answer

```
entity SEQMAJ is
  port (R, I, CLK: in BIT;
        O : out BIT);
end SEQMAJ;

architecture FSM_RTL of SEQMAJ is
  type STATE_TYPE is (S0, S1, S2, S3, S4, S5, S6, S7);
  signal STATE: STATE_TYPE;
begin
```

VHDL Modeling Answer (Continued)

```
-- Process to update state at end of each clock period.
NEXT_STATE: process (R, CLK)
begin
  if (R = '0') then
    STATE <= S0;
  elsif (CLK='1'and CLK'event) then
    case STATE is
      when S0 =>
        if (I = '0') then
          STATE <= S1;
        else
          STATE <= S2;
        end if;
      when S1 =>
        if (I = '0') then
          STATE <= S3;
        else
          STATE <= S4;
        end if;
    end case;
  end if;
end process;
```

17

VHDL Modeling Answer (Continued)

```
      when S2 =>
        if (I = '0') then
          STATE <= S4;
        else
          STATE <= S5;
        end if;
      when S3 =>
        STATE <= S6;
      when S4 =>
        if (I = '0') then
          STATE <= S6;
        else
          STATE <= S7;
        end if;
      when S5 =>
        STATE <= S7;
    end case;
  end if;
end process;
```

18

VHDL Modeling Answer (Continued)

```
when S6 =>
  if (I = '0') then
    STATE <= S1;
  else
    STATE <= S2;
  end if;
when S7 =>
  if (I = '0') then
    STATE <= S1;
  else
    STATE <= S2;
  end if;
end case;
end if;
end process NEXT_STATE;
```

VHDL Modeling Answer (Continued)

```
--
-- Output process
--
OUTPUT: process (STATE)
begin
  case STATE is
    when S7 =>
      O <= '1';
    when others =>
      O <= '0';
  end case;
end process OUTPUT;
end FSMRTL;
```

Implementation Medium Choice

Question:

If the NRE costs for FPGA and CBIC circuits are \$25,000 and \$166,000, respectively, and the cost of individual parts for FPGA and CBIC circuits are \$20 and \$6, respectively, what is the break-even manufacturing volume for these two types of circuits?

Answer:

$$25,000 + 20x = 166,000 + 6x$$

$$14x = 141,000$$

$$x = 10,071$$