Name: KEY **Id#**

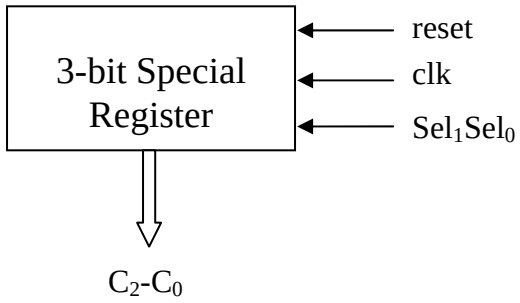
COE 405, Term 031

Design & Modeling of Digital Systems

Quiz# 2

Date: Sunday, Nov. 2, 2003

Q.1. It is required to design a 3-bit register that has the capability to count up, count down, shift left logically, or shift right logically based on a 2-input select. The interface description of the 3-bit register is shown below, where **sel** determines the operation. When sel=00, the register will count up, when sel=01, it will count down, when sel=10, it will shift left logically, and when sel =11, it will shift right logically. The reset is an **asynchronous reset** and the register is **rising-edge** triggered.



(i) Describe an Entity **CS3** for the 3-bit register using type BIT and BIT_VECTOR for the interface signals.

```
Entity CS3 IS
   PORT (reset, clk: IN BIT; sel: IN BIT_VECTOR(1 downto 0); C: OUT
   Bit_Vector(2 Downto 0));
   Constant limit: INTEGER :=7;
END CS3 ;
```

(ii) Model a behavioral Architecture **Behave** for this 3-bit register.

```
Architecture Behave OF CS3 IS
Begin
   Process(clk, reset)
         Variable count: INTEGER := 0;
   Begin
     IF reset = '1' THEN  count := 0;
```

```
ELSIF (clk = '1' AND clk'Event ) THEN
        CASE sel is
            when "00" => count := count+1;
            when "01" => count := count-1;
            when "10" => count := count*2;
            when "11" => count := count/2;
        END case;
        IF (count > limit) Then count := count-limit-1;
        ELSIF count= -1 Then count := limit;
        END IF;
    END IF;
    Case count is
        when 0 => c <= "000";
        when 1 => c <= "001";
        when 2 => c <= "010";
        when 3 => c <= "011";
        when 4 => c <= "100";
        when 5 => c <= "101";
        when 6 => c <= "110";
        when 7 => c <= "111";
        when others => c <= "000";
    End Case;
    END process;
END Behave ;
```

Q.2. Given the following signal assignments, show all transactions placed on each signal. At each event, show transactions that are appended, overwritten, and expired including those occurring at delta time. Show resulting waveforms on each signal.

```
Architecture dataflow of signals IS
    Signal A, B, C, D: Bit := '0';
Begin
  A <= '1' after 10 ns, '0' after 13 ns, '1' after 17 ns, '0' after 25 ns, '1' after 26 ns;
  B <= transport A after 5 ns;
  C <= A after 5 ns;
  D <= Reject 2 ns Inertial A after 5 ns;
End dataflow;
```

		0ns	5ns	10ns	13ns	15ns	17ns	18ns	22ns	25ns	26ns	30ns	31ns
A	Current Value	0	0	1	0	0	1	1	1	0	1	1	1
	Projected Waveform	(1,10) (0,13) (1,17) (0,25) (1,26)	(1,5) (0,8) (1,12) (0,20) (1,21)	(0,3) (1,7) (0,15) (1,16)	(1,4) (0,12) (1,13)	(1,2) (0,10) (1,11)	(0,8) (1,9)	(0,7) (1,8)	(0,3) (1,4)	(1,1)	ϕ	ϕ	ϕ
B	Current Value	0	0	0	0	1	1	0	1	1	1	0	1
	Projected Waveform	(0,5)	ϕ	(1,5)	(1,2) (0,5)	(0,3)	(0,1) (1,5)	(1,4)	ϕ	(0,5)	(0,4) (1,5)	(1,1)	ϕ
C	Current Value	0	0	0	0	0	0	0	1	1	1	1	1
	Projected Waveform	(0,5)	ϕ	(1,5)	~~(1,2)~~ (0,5)	(0,3)	~~(0,1)~~ (1,5)	(1,4)	ϕ	(0,5)	~~(0,4)~~ (1,5)	(1,1)	ϕ
D	Current Value	0	0	0	0	1	1	0	1	1	1	1	1
	Projected Waveform	(0,5)	ϕ	(1,5)	(1,2) (0,5)	(0,3)	(0,1) (1,5)	(1,4)	ϕ	(0,5)	~~(0,4)~~ (1,5)	(1,1)	ϕ