

EE 3921

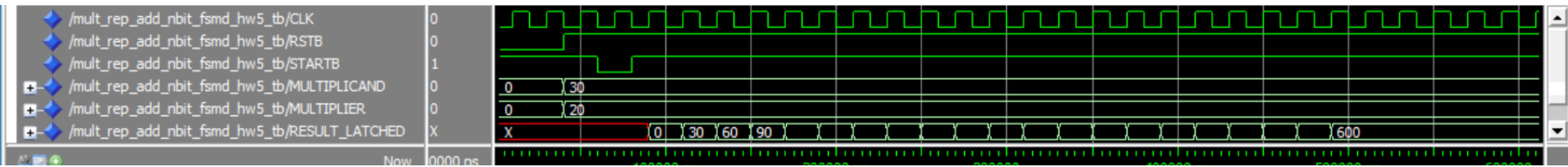
Dr. Johnson

Homework 5

1 – FSMD multiplier

40pts

Modify the repetitive addition multiplier RTL to make it support N bit multiplication. Simulate it for $N = 5$ (multiply 20 by 30) (code and plot – do not submit the testbench)



2 – Determine the value of `data_out` at the end of the following sequence of `ctrl` and `data_in` inputs. Assume 1 `ctrl` and `data_in` input per clock and that both are stable on the rising clock edge. – show your work – do not simulate 30pts

```

library ieee;
use ieee.std_logic_1164.all;

entity hw5 is
  generic(N: integer := 8);
  port(
    i_clk:      in std_logic;
    i_rstb:     in std_logic;
    i_ctrl:     in std_logic_vector(1 downto 0);
    i_data_in:  in std_logic_vector(N-1 downto 0);
    o_data_out: out std_logic_vector(N-1 downto 0)
  );
end;

architecture behavioral of hw5 is
  --
  -- Signals
  --
  signal reg:      std_logic_vector(N-1 downto 0);
  signal reg_next: std_logic_vector(N-1 downto 0);

begin
  --
  -- Control Logic
  --
  process(all)
  begin
    case i_ctrl is
      when "00" => reg_next <= reg;
      when "01" => reg_next <= reg(N-2 downto 0) & i_data_in(0);
      when "10" => reg_next <= i_data_in(N-1) & reg(N-1 downto 1);
      when others => reg_next <= NOT reg;
    end case;
  end process;
end;

```

```

-- Synchronous Update Section
--
process(i_clk,i_rstb)
begin
  if (rising_edge(i_clk) AND (i_rstb = '1')) then
    reg <= reg_next;
  end if;
end process;

--
-- Output Section
--
o_data_out <= reg;

end;

```

| Reg | Data in | reg next |
|-----------|-----------|-----------|
| uuuu uuuu | 1001 1001 | uuuu uu1 |
| uuuu uu1 | 0010 0010 | 0uuu uuuu |
| 0uuu uuuu | 0011 0011 | 1uuu uuuu |
| 1uuu uuuu | 0100 0100 | 01uu uuuu |
| 01uu uuuu | 0101 0101 | 011u uuuu |
| 011u uuuu | 0110 0110 | 01uu uu1 |
| 01uu uu1 | 0111 0111 | 01uu uu1 |
| 01uu uu1 | 1000 1000 | 001u uuuu |
| 001u uuuu | 0001 0001 | 0001 uuuu |

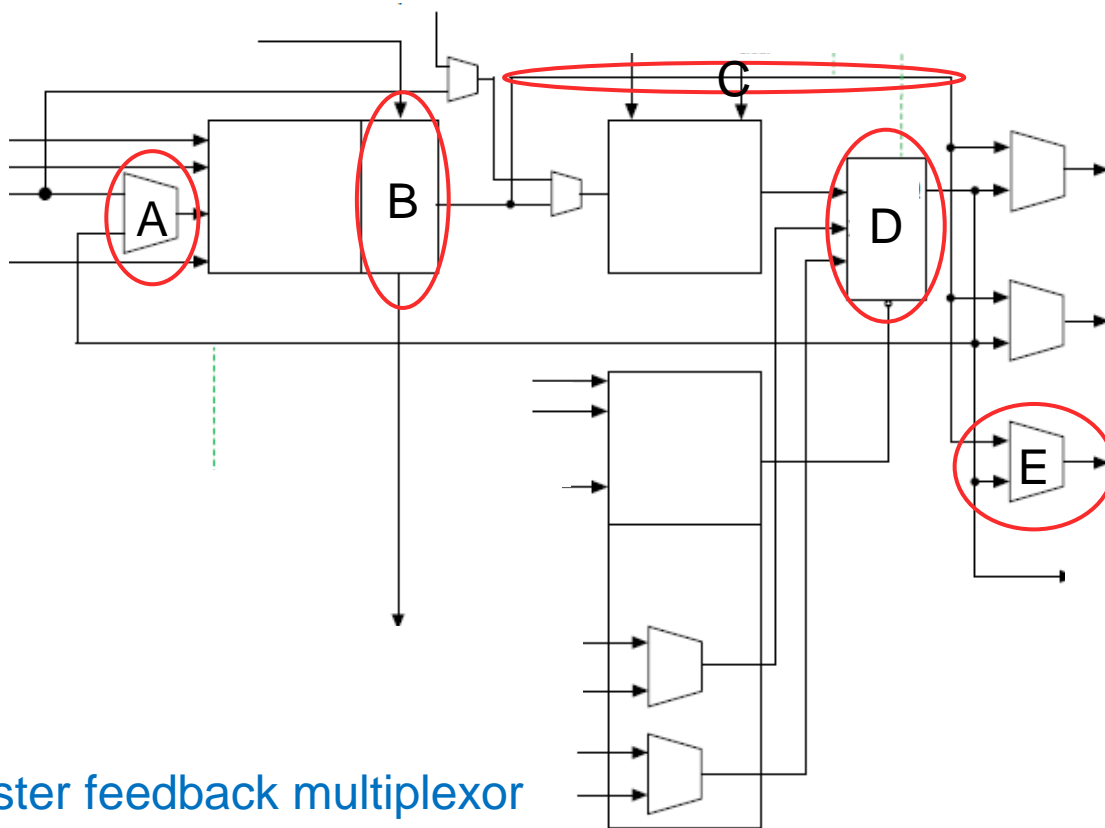
| data_in | ctrl |
|---------|--------|
| hex | binary |
| 0x99 | 01 |
| 0x22 | 10 |
| 0x33 | 11 |
| 0x44 | 00 |
| 0x55 | 10 |
| 0x66 | 10 |
| 0x77 | 01 |
| 0x88 | 00 |
| 0x11 | 10 |

Data_out

001u uuuu

3 – Identify the purpose of each highlighted block

10pts



A: Register feedback multiplexor

B: Carry Chain

C: Register Bypass

D: Register

E: Local routing multiplexor

4 – Provide the hex value for the LUT Mask of a 4 input LUT with signals A,B,C,D tied to inputs 1,2,3,4 respectively when the compiled logic represents $out = ((A+B) + (C\&D))$ 20pts

| Inputs | | | | E | | | F | OUT |
|--------|---|---|---|-----|-----|-------|-----|-----|
| A | B | C | D | A+B | C D | E + F | | |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | b0 | |
| 0 | 0 | 0 | 1 | 0 | 0 | 0 | b1 | |
| 0 | 0 | 1 | 0 | 0 | 0 | 0 | b2 | |
| 0 | 0 | 1 | 1 | 0 | 1 | 1 | b3 | |
| 0 | 1 | 0 | 0 | 1 | 0 | 1 | b4 | |
| 0 | 1 | 0 | 1 | 1 | 0 | 1 | b5 | |
| 0 | 1 | 1 | 0 | 1 | 0 | 1 | b6 | |
| 0 | 1 | 1 | 1 | 1 | 0 | 1 | b7 | |
| 1 | 0 | 0 | 0 | 0 | 0 | 0 | b8 | |
| 1 | 0 | 0 | 1 | 1 | 0 | 1 | b9 | |
| 1 | 0 | 1 | 0 | 1 | 0 | 1 | b10 | |
| 1 | 0 | 1 | 1 | 1 | 0 | 1 | b11 | |
| 1 | 1 | 0 | 0 | 0 | 1 | 1 | b12 | |
| 1 | 1 | 0 | 1 | 1 | 1 | 1 | b13 | |
| 1 | 1 | 1 | 0 | 1 | 1 | 1 | b14 | |
| 1 | 1 | 1 | 1 | 1 | 1 | 1 | b15 | |

HEX

0xFEEE

0xFFF8 if you did A B C D in truth table