1. (7 points) Draw the transistor-level diagram of a CMOS two-input NOR gate.

![NOR Gate Diagram](image)

2. (10 points) Write a VHDL entity (3 points) and architecture (7 points) of a two-input OR gate with the generics, TPLH and TPHL, which reflect the time for the output to make a low to high or high to low transition, respectively.

```vhdl
library ieee;
use ieee.std_logic_1164.all;

entity OR_DEL is
  generic (TPLH, TPHL : time := 2 ns);
  port (x, y : in std_logic;
          f : out std_logic);
end OR_DEL;

architecture BEHAV of OR_DEL is
begin
  process (x,y)
  variable current, old : std_logic;
  begin
    current := x or y;
    f <= x or y;
    if (current = '1' and old = '0') then
      f <= x or y after TPHL;
    elsif (current = '0' and old = '1') then
      f <= x or y after TPLH;
    end if;
    old := current;
  end process;
end BEHAV;
```

3. (1 point) **Routing** is the process of making the connections between standard cells.
4. (5 points) If the NRE costs for FPGA and CBIC circuits are $21,000 and $187,000, respectively, and
the cost of individual parts for FPGA and CBIC circuits are $15 and $7, respectively, what is the break-
even manufacturing volume for these two types of circuits?

\[ x - \text{number of units} \quad 21000 + 15x = 187000 + 7x, \quad 8x = 166000, \quad x = 20750 \]

5. (5 points) What kind of hardware element will be inferred by a synthesis tool from the
following model? A flip-flop (because of the edge behavior) with synchronous reset.

```vhdl
library ieee;
use ieee.std_logic_1164.all;

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 (CLK'event and CLK = '1') then
                if (RESET = '1') then
                    Z <= '0';
                else
                    Z <= A nor B;
                end if;
            end if;
        end process;
    end EXAMPE;
end EXAMPLE;
```

6. (10 points) For the data lifetime chart shown, use the left edge algorithm to obtain an efficient
register allocation.

```
<table>
<thead>
<tr>
<th></th>
<th>A</th>
<th>B</th>
<th>C</th>
<th>D</th>
<th>E</th>
<th>F</th>
<th>G</th>
<th>H</th>
<th>I</th>
<th>J</th>
<th>K</th>
<th>L</th>
</tr>
</thead>
<tbody>
<tr>
<td>S1</td>
<td></td>
<td>X</td>
<td></td>
<td>X</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>S2</td>
<td>X</td>
<td>X</td>
<td></td>
<td></td>
<td>X</td>
<td>X</td>
<td>X</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>S3</td>
<td>X</td>
<td></td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>S4</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>S5</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td></td>
</tr>
<tr>
<td>S6</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td></td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>S7</td>
<td></td>
<td>X</td>
<td></td>
<td>X</td>
<td></td>
<td>X</td>
<td>X</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>X</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th></th>
<th>D</th>
<th>C</th>
<th>G</th>
<th>J</th>
<th>B</th>
<th>F</th>
<th>K</th>
<th>I</th>
<th>A</th>
<th>L</th>
<th>H</th>
<th>E</th>
</tr>
</thead>
<tbody>
<tr>
<td>S1</td>
<td>X</td>
<td>X</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>S2</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>S3</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>S4</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>S5</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>S6</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>S7</td>
<td></td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
```
7. (1 point) Typically, a ___test bench___ is developed to validate a VHDL behavioral model.

8. (1 point) A(n) _SDF file_ provides the gate-level circuit with accurate timing backannotated from the layout.

9. (1 point) ___Pragmas_ are inserted into VHDL models to give instructions to synthesis or other tools.

10. (18 points) Create a VHDL entity named _en_dec_328 that represents a 3-to-8 decoder with an active-low enable input which has an architecture which uses a case statement to represent the functionality of the decoder. Create a second entity and its accompanying architecture that represents a 4-to-16 decoder by using two instances of the _en_dec_328_ entity.

```
library ieee;
use ieee.std_logic_1164.all;

entity EN_DEC_3TO8 is
    port (EN : in std_logic;
          I : in std_logic_vector(2 downto 0);
          O : out std_logic_vector(7 downto 0));
end EN_DEC_3TO8;

architecture BEHAV of EN_DEC_3TO8 is
begin
    process(EN, I)
    begin
        case EN is
        when '0' =>
            case I is
            when "000" => O <= "00000001";
            when "001" => O <= "00000010";
            when "010" => O <= "00000100";
            when "011" => O <= "00001000";
            when "100" => O <= "00010000";
            when "101" => O <= "00100000";
            when "110" => O <= "01000000";
            when "111" => O <= "10000000";
            when others => O <= "00000000";
            end case;
        end case;
    end case;
end process;
end BEHAV;
```

```
use ieee.std_logic_1164.all;

entity DEC_4TO16 is
  port (I : in std_logic_vector(3 downto 0);
        O : out std_logic_vector(15 downto 0));
end DEC_4TO16;

architecture STRUCT of DEC_4TO16 is
  signal I3BAR : std_logic;
  component EN_DEC_3TO8C
    port (EN : in std_logic;
          I : in std_logic_vector(2 downto 0);
          O : out std_logic_vector(7 downto 0));
  end component;
  for all : EN_DEC_3TO8C use entity work.EN_DEC_3TO8(BEHAV);
begin
  I3BAR <= not I(3);
  U1 : EN_DEC_3TO8C
    port map (I(3), I(2 downto 0), O(7 downto 0));
  U2 : EN_DEC_3TO8C
    port map(I3BAR, I(2 downto 0), O(15 downto 8));
end;

11. (15 points) Modify the following VHDL model to use block(s) instead of processes.

library ieee;
use ieee.std_logic_1164.all;

entity BUFF_REG is
  generic (STRB_DEL, EN_DEL, ODEL: TIME);
  port (DI: in std_logic_vector (1 to 8);
        DS1, NDS2, STRB : in std_logic;
        DO: out std_logic_vector (1 to 8));
end BUFF_REG;

architecture THREE_PROC of BUFF_REG is
  signal REG : std_logic_vector (1 to 8);
  signal ENBLD : std_logic;
begin
  PREG: process (STRB)
  begin
    if (STRB = '1') then
      REG <= DI after STRB_DEL;
    end if;
  end process PREG;

  ENABLE : process (DS1, NDS2)
  begin
    ENBLD <= DS1 and not NDS2 after EN_DEL;
  end process ENABLE;

  OUTPUT : process (REG, ENBLD)
  begin
    if (ENBLD = '1') then
      DO <= REG after ODEL;
    else
      DO <= "ZZZZZZZZ" after ODEL;
    end if;
  end process OUTPUT;
end THREE_PROC;

begin
  PREG: block (STRB = '1')
  REG <= guarded DI after STRB_DEL;
end PREG;

  ENABLE: block
  ENBLD <= DS1 and not NDS2 after EN_DEL;
end ENABLE;

  OUTPUT: block
  DO <= REG after ODEL when ENBLD = '1' else "ZZZZZZZZ" after ODEL;
end OUTPUT;
Consider the following VHDL code:

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

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

12. (14 points) The following tasks refer to the VHDL code above. Assume that there are no hardware constraints.

a. (7 points) Derive an ASAP schedule.

<table>
<thead>
<tr>
<th>Ready</th>
<th>Schedule</th>
</tr>
</thead>
<tbody>
<tr>
<td>{1, 2, 3, 4, 5, 6}</td>
<td>{1, 2, 3, 4, 5, 6}</td>
</tr>
<tr>
<td>{7, 8}</td>
<td>{7, 8}</td>
</tr>
<tr>
<td>{9, 10}</td>
<td>{9, 10}</td>
</tr>
<tr>
<td>{11, 12, 13}</td>
<td>{11, 12, 13}</td>
</tr>
</tbody>
</table>

b. (7 points) Derive an ALAP schedule.

<table>
<thead>
<tr>
<th>Ready</th>
<th>Schedule</th>
</tr>
</thead>
<tbody>
<tr>
<td>{11, 12, 13}</td>
<td>{11, 12, 13}</td>
</tr>
<tr>
<td>{2, 6, 9, 10}</td>
<td>{2, 6, 9, 10}</td>
</tr>
<tr>
<td>{1, 4, 7, 8}</td>
<td>{1, 4, 7, 8}</td>
</tr>
<tr>
<td>{3, 5}</td>
<td>{3, 5}</td>
</tr>
</tbody>
</table>
13. (10 points) Derive a list schedule using the critical path priority metric for the VHDL code above, using the following hardware constraint; all operations are done in an ALU module and there are two ALU modules available.

**Solution:**

<table>
<thead>
<tr>
<th>Op</th>
<th>Length</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>3</td>
</tr>
<tr>
<td>2</td>
<td>2</td>
</tr>
<tr>
<td>3</td>
<td>4</td>
</tr>
<tr>
<td>4</td>
<td>3</td>
</tr>
<tr>
<td>5</td>
<td>4</td>
</tr>
<tr>
<td>6</td>
<td>2</td>
</tr>
<tr>
<td>7</td>
<td>3</td>
</tr>
<tr>
<td>8</td>
<td>3</td>
</tr>
<tr>
<td>9</td>
<td>2</td>
</tr>
<tr>
<td>10</td>
<td>2</td>
</tr>
<tr>
<td>11</td>
<td>1</td>
</tr>
<tr>
<td>12</td>
<td>1</td>
</tr>
<tr>
<td>13</td>
<td>1</td>
</tr>
</tbody>
</table>

List \{3, 5, 1, 4, 7, 8, 2, 6, 9, 10, 11, 12, 13\}  
Ready Schedule  
\{1, 2, 3, 4, 5, 6\} \{3, 5\}  
\{1, 2, 4, 6, 7, 8\} \{1, 4\}  
\{2, 6, 7, 8\} \{7, 8\}  
\{2, 6, 9, 10\} \{2, 6\}  
\{9, 10\} \{9, 10\}  
\{11, 12, 13\} \{11, 12\}  
\{13\} \{13\}

14. (1 point) __ Allocation, scheduling, compilation___ is one algorithmic-level synthesis task.

15. (1 point) _____ Communication____ is the hardest problem.