1. (16 points) Find the simplest sum-of-products equivalent to the expression below using Boolean algebra only.

\[
(A'B'D' + BCD' + BC'D' + AC')' = (A+B+D)(B'+C'+D)(B'+C+D) (A'+C)
\]

\[
= (AB' + AC' + BC' + D)(A'B' + C + A'D)
\]

\[
= (AB'C + A'BC'D + A'B'D + CD + A'D)
\]

Show how your simplified version of the above expression can be implemented using NAND gates and inverters only (you need not limit yourself to simple gates).
Draw a CMOS two input NAND gate using NFETs and PFETs. Show how to extend this to a three input NAND gate using two more transistors.

If the NAND circuit you drew above is implemented using CMOS NAND gates, how many transistors does it require?

24
2. (6 points) Fill in the Karnaugh map below so that it corresponds to the function 
\( F(A,B,C,D) = \Sigma m(1,2,6,8,10,14), \ d(A,B,C,D) = \Sigma m(0,4,7,9,13) \). Derive a product of sums expression for this function, taking full advantage of the don’t care conditions.

Covering the 0’s gives us \( F’ = BC’ + CD \), so \( F = (B’ + C)(C’ + D’) \)
3. (10 points) Draw a logic diagram that directly corresponds to the VHDL module shown below. Use simple gates only. Every signal that appears in the VHDL should be labeled in your logic diagram.

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

entity foo is port (
    A : in std_logic;
    B : in std_logic_vector(3 downto 0);
    X : out std_logic_vector(3 downto 0),
    Y : out std_logic);
end foo;
architecture a1 of foo is
begin
    signal Z : std_logic_vector(4 downto 0);
    process (A, B, Z) begin
        Z(4 downto 1) <= B; Z(0) <= A;
        for i in 0 to 3 loop
            if i = 0 or i = 2 then
                Z(i+1) <= B(i) and Z(i);
            elsif i = 1 then
                Z(i+1) <= B(i) or Z(i);
            end if;
        end loop;
        X <= B xor Z(3 downto 0);
        Y <= Z(4);
    end process;
end a1;
```

```
A=Z(0)

B(0)  \rightarrow  X(0)

B(1)  \rightarrow  X(1)
  \rightarrow  Z(1)

B(2)  \rightarrow  X(2)
  \rightarrow  Z(2)

B(3)  \rightarrow  X(3)
  \rightarrow  Z(3)
  \rightarrow  Z(4)=Y
```
4. (15 points) With respect to the circuit shown below, give a general expression for output $X_i$ in terms of inputs $B, A_0, \ldots, A_i$.

\[ X_i = A_i (B \oplus A_0 \oplus \cdots \oplus A_{i-1}) \]

Suppose this circuit is extended to 64 bits and the exclusive-or gates are implemented using NAND gates. What is the worst-case delay from the time that an input changes until the time an output changes, assuming 1 ns of delay per gate.

\[ 3 \times 64 = 192 \text{ ns} \]
Complete the schematic diagram shown below so that it implements a lookahead version of the circuit.

What is the worst-case delay for a 64 bit version of the lookahead circuit?

$3 \times 7 = 21 \text{ ns}$
5. (15 points) The simulation output shows a program executing on the dual processor system from design problem 2. Fill in the blanks below with the information that belongs in each of the blanked out areas in the simulation output.

- A. \textit{branchZero} 
- B. \textit{1FEC} 
- C. \textit{1FEC} 
- D. \textit{mLoad} 
- E. \textit{0006} 
- F. \textit{FFEC} 
- G. \textit{FFF4} 
- H. \textit{loadPID} 
- I. \textit{0001} 
- J. \textit{negate}
6. (12 points) For the sequential circuit shown below, complete the timing diagram to show when the input must be stable and when the output can be changing. For the input, label the start and end of each stable period relative to the rising clock edge (so write \(-3\) to indicate 3 ns before the clock edge, for example). Similarly label the ends of the period when the output can be changing. Assume that the flip flop setup time is 1.5 ns, the flip flop propagation delay is between 1.5 and 2.5 ns, the hold time is 1 ns and all gates have delays that vary between .4 ns and 1.3 ns.

What is the smallest clock skew for which it’s possible for the circuit to experience internal hold-time violations?

.9 ns

For a clock skew of 1 ns, what is the largest clock period for which the circuit can experience internal setup time violations?

8.9 ns
7. (10 points) Complete the state table shown below, so that it corresponds to the state diagram shown at left.

Write an output equation for $X$. You may use the don’t care conditions to simplify the expression.

$$S_1'B' + S_0'A + S_0A'B$$
8. (8 points) Show how to configure the PAL shown below to implement the logic equations.

\[ X = A'D + BCD' + AC' \quad Y = A'B' + BC + C'D \quad Z = A'D + BCD' + AC' \]

Do not simplify the equations first.

If you were to implement the same equation using a PLA, what is the smallest number of product terms that would be required for the PLA (again, assuming that the equations are not simplified first)?

*6 terms are enough.*
9. (12 points) The table shown below represents a 2-way set associative cache (combined instruction and data cache). The ‘<’ and ‘>’ symbols in the center point toward the least recently used of the two entries in a given row. So for example, the right entry in row 4 has been used more recently than the left entry.

| tag | data | | tag | data |
|-----|------| |-----|------|
| 0   | 0a4  | <   | 0   | 0a5  |
| 1   | b23  | <   | 1   | 0a5  |
| 2   | 0a4  | >   | 2   | 0a5  |
| 3   | 0a4  | >   | 3   | 20b  |
| 4   | 20c  | <   | 4   | 20b  |
| 5   | 20c  | <   | 5   | 102  |
| 6   | 302  | >   | 6   | b23  |
| 7   | c25  | >   | 7   | 1fc  |
| 8   | b23  | >   | 8   | 0a5  |
| 9   | acb  | <   | 9   | 20b  |
| a   | 07b  | >   | a   | 20b  |
| b   | 20c  | >   | b   | 20b  |
| c   | 0a3  | <   | c   | 20b  |
| d   | 0c2  | >   | d   | 0a4  |
| e   | 20c  | <   | e   | 0a3  |
| f   | 0a4  | >   | f   | 0a5  |

Show how the state of the cache changes, when the processor fetches and executes the two instructions shown below, assuming that a new value replaces the least recently used entry. Assume that cache entries are selected using the low order 4 bits of the memory address and that the tag consists of the high order 12 bits. Be sure to update the ‘<’ and ‘>’ symbols where appropriate.

<table>
<thead>
<tr>
<th>address</th>
<th>data</th>
</tr>
</thead>
<tbody>
<tr>
<td>b236</td>
<td>1002</td>
</tr>
<tr>
<td>b237</td>
<td>ac2d</td>
</tr>
</tbody>
</table>

What is the value in the accumulator, after the above two instruction complete?

25c7

What is the value of the instruction register after the next instruction executes.

4738
10. (20 points) The VHDL module defined below looks for the longest string of consecutive 1s in the input and outputs the length of this longest string. More specifically, the result output is the number of 1s in the longest string seen so far (since reset went low). Complete the block diagram/schematic on the following page so that it corresponds to the circuit specified by the VHDL module. Do this by adding wires and simple gates only.

```vhdl
entity maxRunner is port (  
    clk, reset, A : in std_logic;  
    result: out std_logic_vector(3 downto 0));  
end maxRunner;
architecture a1 of maxRunner is  
signal run, maxRun: std_logic_vector(3 downto 0);  
begin  
    process (clk) begin  
        if clk'event and clk = '1' then  
            if reset = '1' then  
                run <= "0000"; maxRun <= "0000";  
            elsif A = '0' then  
                run <= "0000";  
            else  
                run <= run + '1';  
                if run = maxRun then  
                    maxRun <= run + '1';  
                end if;  
            end if;  
        end if;  
    end process;  
    result <= maxRun;  
end a1;
```
What is the smallest clock period for which this circuit is guaranteed to have no internal setup time violations. Assume that all gates have a 1 ns delay, that flip flops have a setup time of 1.5 ns and a propagation delay of 2.5 ns and that there is zero clock skew. Also assume that the equality comparison circuits are implemented using exnor gates with outputs and-ed together and that the increment circuit is implemented using ripple-carry.

The increment circuit has a delay of 4 ns due to the carry chain. The comparison circuit has a delay of 2 ns. So, the largest combinational circuit delay from a flip flop output to a flip flop input passes through the increment circuit and then through three gates that precede the maxRun register. So, the largest combinational circuit delay is 7 ns (4 for the increment circuit, and 3 for the gates). Adding this to the flip flop setup time and propagation delay gives a total of 11 ns, as the minimum clock period.