1. (10 points) Define the term *combinational circuit*.

Let \( x \) be a signal and \( p \) be a purely asynchronous process within a VHDL architecture. Under what conditions is it *necessary* for \( x \) to be included in the sensitivity list for \( p \)?

Write a short VHDL process involving *std_logic* signals \( a \), \( b \) and \( c \) that will cause a circuit synthesizer to generate a latch.

How does a flip flop behave when it is metastable?

What is the maximum time period for which a flip flop can be metastable.
2. (10 points). Use the Karnaugh map below to find a minimum sum-of-products expression for $\Sigma_m(0,1,3,4,5,8,9,12,14)$. How many simple gates of each type are needed to implement this expression (without further simplification)? How many LUT4s?

<table>
<thead>
<tr>
<th>CD</th>
<th>00</th>
<th>01</th>
<th>11</th>
<th>10</th>
</tr>
</thead>
<tbody>
<tr>
<td>00</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>01</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>A B</td>
<td>01</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>11</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>10</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Use the Karnaugh map below to find a minimum *product-of-sums* expression for $\Sigma_m(0,1,9,13,15), \Sigma_d(3,4,5,6,8)$. Make full use of the don’t cares.

<table>
<thead>
<tr>
<th>CD</th>
<th>00</th>
<th>01</th>
<th>11</th>
<th>10</th>
</tr>
</thead>
<tbody>
<tr>
<td>00</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>01</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>A B</td>
<td>01</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>11</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>10</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
3. (10 points) The diagram below shows a generic state machine. Assume that the state is encoded using 5 bits.

Does this circuit violate the hold time constraint?

If so, how could you eliminate the violations? If not, what is the smallest clock skew for which hold time violations are possible.

What is the smallest safe clock period for this circuit (without any modifications to handle hold time violations)?

If the clock goes from low to high at time $t_0$, during what time period must the inputs be stable?

Suppose that two copies of this circuit are instantiated together, with the output of one connected to the input of the other. For this combined circuit, what is the minimum safe clock period?
4. (10 points) The processor simulation below includes several labeled blanks. Fill in the correct values in the spaces below.

A. ________________  B. ________________  C. ________________
D. ________________  E. ________________

The instruction set for the WASHU-2 processor appears below.

0000  **halt**  1xxx  constant load
0001  **negate**  2xxx  direct load
01xx  **branch**  3xxx  indirect load
02xx  **branch if zero**  5xxx  direct store
03xx  **branch if positive**  6xxx  indirect store
04xx  **branch if negative**  8xxx  **add**
05xx  **indirect branch**  cxxx  **and**
5. (15 points) Write a program in the WASHU-2 assembly language that implements the following C-style pseudo-code. Note that the lines defining i, j and p are included below. For each of the branch instructions in your program, also show the actual machine instruction that is generated by the assembler for that instruction. Write these to the right of the assembler instructions.

```c
while (i != 0) {
    if (j > i) *p = *p + j;
    i--;
}
```

```
location 0100
```

```
location 0120
```

```
i:  20
j:  7
p:  0123
```
6. (10 points) Consider an SRAM with 2K words of 32 bits each. How many address bits are needed to address these words? (Reminder \(1024 = 2^{10}, 4096 = 2^{12}, 16384 = 2^{14}, 65536 = 2^{16}\).)

Assuming that the central memory array has the same number of rows as it has columns, how many rows are there?

How many words are stored in each row?

How many of the address bits are used by the row decoder?

How many are used by the column decoder?

Draw a diagram of the basic storage cell that is typically used in an SRAM.
7. (10 points) Consider a version of the WASHU-2 processor that is equipped with a 2-way set-associative cache with 256 rows (so, 512 words altogether) for both instructions and data, with the partial contents shown below. The least recently used (LRU) bit in the center column is equal to 0 if the left-hand entry has been least recently used and is equal to 1 if the right-hand entry has been least recently used.

<table>
<thead>
<tr>
<th>tag</th>
<th>value</th>
<th>LRU</th>
<th>tag</th>
<th>value</th>
</tr>
</thead>
<tbody>
<tr>
<td>49</td>
<td>22</td>
<td>0003</td>
<td>0</td>
<td>82</td>
</tr>
<tr>
<td>4a</td>
<td>34</td>
<td>c789</td>
<td>1</td>
<td>24</td>
</tr>
<tr>
<td>4b</td>
<td>23</td>
<td>344a</td>
<td>1</td>
<td>71</td>
</tr>
<tr>
<td>4c</td>
<td>20</td>
<td>1b4c</td>
<td>0</td>
<td>50</td>
</tr>
</tbody>
</table>

If PC=244a, what instruction is fetched next?

Assume ACC=5 initially. What value is in the ACC after the next instruction executes?

Show how the fetching and execution of this instruction modifies the cache by marking the changes in the table above.

Suppose that the instruction at location 244b is 504c. Show how the fetching and execution of this instruction changes the cache contents, by marking your changes on the diagram above (be sure to update the LRU bits, if appropriate).

What value is in the ACC after this instruction executes?
8. (10 points) On a modern processor used in a laptop, approximately how long does it take to retrieve an instruction from the L1 instruction cache? How long does it take to retrieve an instruction from main memory, if it’s not present in any of the caches?

What is the advantage of having multiple general purpose registers in processor (as opposed to the single accumulator used in the WASHU-2)?

List two instructions that could be added to the WASHU-2 instruction set that would improve the performance of typical programs. For each instruction give an instruction summary similar to the one shown below for the brZero instruction. Your new instructions should be natural additions to the existing instruction set. In particular, do not re-use any instruction codes already assigned.

02xx brZero if ACC=0 then PC=PC+ssxx

Explain how these instructions improve the performance of programs running on the processor. Estimate the amount by which they would improve the performance of programs using them.
9. (10 points) Consider the VHDL process shown below. Assume that all signals are 16 bits.

```vhdl
[1] process (clk, a, b, c) begin
[2]     x <= a + b;
[3]     if rising_edge(clk) then
[4]         if a = c then
[5]             y <= b + c; z <= c;
[6]         elsif b > c then
[7]             y <= a - x;
[8]     end if; end if; end process;
```

Suppose that a circuit is synthesized for this process for the FPGA on our prototype board. How many flip flops are used by this circuit? How many latches?

Approximately how many LUT4s are used to evaluate the condition in the if-statement on line 4?

Approximately how many LUT4s are used to evaluate the condition in the elsif-statement on line 6?

Approximately how many LUT4s are used by all the assignments to y?