1. Consider the pseudo-C code below.

```c
Inputs: uint16_t A, uint16_t B, bit hold
Outputs: uint32_t Y, bit done
Storage: uint32_t tmp
main()
while(1){
    while(hold);
    done = 0;
    tmp = A;
    while(B != 0){
        if (B(0) == 1){
            Y += tmp;
        }
        tmp = tmp << 1;
        B = B >> 1 ;
    }
    done = 1;
}
```

a. What does this code implement?
b. Draw an HLSM to represent the system.
c. Draw the datapath for this system.
d. Connect the datapath to the control unit. Label all signals between them.
2. Design a system that determines whether the given unsigned 8-bit number A is prime. The system starts with pushing the button S, and activates the output E when finished the computation. Output Y determines if A is prime. You may assume that you have a combinational divider unit that gets two inputs A1 and A2, and outputs the remainder R and quotient Q in 3 clock cycles.

a. Draw an HLSM to represent the system.
b. Draw the datapath for this system.
c. Connect the datapath to the control unit. Label all signals between them.
3. Consider the entire system of datapath and control flow as follows. All numbers are in signed 2’s complement format. The delay of all components are given in the table below. Find the maximum frequency. Assume no hold-time violations and any subtractor/adder has a ripple-carry structure, made up of only Half- or Full-adders.

<table>
<thead>
<tr>
<th>Component</th>
<th>Delay (ps)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Inverter (NOT)</td>
<td>5</td>
</tr>
<tr>
<td>Gate (other than NOT)</td>
<td>15</td>
</tr>
<tr>
<td>Half Adder</td>
<td>25</td>
</tr>
<tr>
<td>Full Adder</td>
<td>35</td>
</tr>
<tr>
<td>2:1 Mux</td>
<td>30</td>
</tr>
<tr>
<td>DFF&lt;sub&gt;Set&lt;/sub&gt;</td>
<td>25</td>
</tr>
<tr>
<td>DFF&lt;sub&gt;clock-q&lt;/sub&gt;</td>
<td>50</td>
</tr>
<tr>
<td>Memory (read)</td>
<td>100</td>
</tr>
</tbody>
</table>
4. Design a system which reorders the data using a bubble sort algorithm with the smallest entry at memory index 0. Assume a memory block contains 256 entries. Assume your memory has two read ports and can write two entries at a time.

   a. Draw an HLSM to represent the system.
   b. Draw the datapath for this system
   c. Connect control and datapath
   d. Draw the FSM and label signals controlling transitions between the states
   e. Implement the MSB of FSM