HW11

1pt
1a) \(2^4 = 16\)  \(2^4 - 1\) states in an LFSR is OK too

1pt
1b) stays in state 0000

2pts
1c) 0001, 1000, 0100, 0010, 1001, 1100, 0110, 1011, 0101, 1101, 1110, 0011, 1010, 1111, 0111, 0011 \(\Rightarrow\) repeats

2pts
1d) \(2^{16} - 1 = 2^{14} \cdot 2^{30}\) states

1GHz \(\Rightarrow\) \(10^9 = 2^{30}\) states/sec

Sequence repeats \(\leq 2^{34}\) seconds \(\Rightarrow\) repeats in 29 years

1 year = \(10 \times 10^3 = 2^5 \cdot 2^{20}\) so \(\approx 1\) century

2pts
2a)

2pts
2b)

\[
Q_i \oplus \prod_{j=0}^{i-1} Q_j
\]

in general,
3a)  

<table>
<thead>
<tr>
<th>Add</th>
<th>500</th>
<th>501</th>
<th>502</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>5</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>B</td>
<td>5</td>
<td>1</td>
<td>2</td>
</tr>
<tr>
<td>C</td>
<td>10</td>
<td>1</td>
<td>-1</td>
</tr>
</tbody>
</table>

8 pts

3b) \( x \) is in R6 because inst 500 is R6 = R6 + R6  
\( \iff x = x + x \)  
\( i \) is in R3  
inst 501 is R3 = R3 + R4  
\( \iff i = i + 1 \)

2 pts

3c) R4 and R5 hold important loop variables  
R4 = 1 is the increment  
R5 = 2 is the limit/compare  
(grade 1 anything that shows they understand what is going on in part b) 3b and 3c

3d) NOT GRADED  
The ALU would typically have an output NC, V, Z where  
NC = 1 if the result is negative  
Z = 1 if the result is zero  
Using your loadable counter from part 2b  
Need an offset in instruction  
to add (sub) to  
current PC

Diagram: [Logic, Program Counter, Memory, Instruction Decode, Register File, ALU, Control Logic]
2 pts  a) IDLE, INIT, WAIT, RUNNING
2 pts  b) GO, READY
2 pts  c) GREEN, ON, START
2 pts  d) GO
4 pts  e) Q, Q̅

Not required

<table>
<thead>
<tr>
<th>States</th>
<th>GREEN</th>
<th>ON</th>
<th>START</th>
</tr>
</thead>
<tbody>
<tr>
<td>IDLE</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>INIT</td>
<td>0</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>WAIT</td>
<td>0</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>RUNNING</td>
<td>1</td>
<td>1</td>
<td>0</td>
</tr>
</tbody>
</table>

1 pt for getting:
- 4 states
- outputs in states
- arrows for transitions

1 pt for getting it all right
5)\[\begin{array}{c}
\begin{array}{c}
A \\
B \\
C
\end{array}
\end{array}\]
\[\begin{array}{c}
F \\
G
\end{array}\]

2 pts

8 3, 8 reg: \(\begin{array}{c}
\text{Din} \\
\text{DQ} \\
\text{DQ} \\
\text{DQ} \\
\text{DQ} \\
\text{DQ} \\
\text{DQ} \\
\text{Dout}
\end{array}\)

2 pts

6) each 4-wire takes 6 bits \(4 \times 4 \times 6 \text{ bits} = 96 \text{ bits}\)

1 pt

b) yes

2 pts

c) no. EX: \(E_0 - W_3, E_1 - W_2, E_2 - W, E_3 - W_0\)

any corner-to-corner knocks out a lot of options

any up/down or left/right inversion

1 pt for "no"

1 pt for example
Once $W_0 - E_3$ and $W_1 - E_2$ are connected, you can't connect $W_2$ or $W_3$ to $E_0$ or $E_1$. 