Intro to Digital Logic, Lab 7
Useful Components

Lab Objectives
Over the last 6 labs we’ve learned how to do most kinds of basic logic, but there are some standard elements that tend to come up over and over again. This lab will help you get some experience with them now, and add them to your toolkit in advance of the final project.

Design Problem – CyberWar
In the last lab we built a simple Tug of War game, and by now you’ve already crushed your room-mate into submission. Now it’s the hardware’s turn. Your goal is to develop a computer opponent to play against, as well as a scorekeeper that can show exactly how badly it beats you…

Counters
First off, take your lab #6 and replace the “winner” system with counters. Specifically, develop a 3-bit counter (holds values 0..7). It starts at 0, and whenever a “win” comes in to it, it increments its current value by 1. This is a simple FSM. Note that we assume once one player gets to 7 the game is over, so it doesn’t matter what happens when a player with 7 points gets one more.

Now, alter your lab #6 so that there is a counter for each player, which drives a per-player 7-segment display with the current score for that player. Whenever someone wins, you increment the appropriate player’s score, then restart the game (i.e. automatically reset the playfield). Resetting the entire game will reset the playfield and score, while winning only resets the playfield.

LFSRs
To build a cyber-player, we need to create a random number generator to simulate the button presses. In hardware, the simplest way to do this is generally an LFSR (linear feedback shift register). It consists of a set of N D-flip-flops (DFF1…DFFN), where the output of DFFi is the input of DFFi+1. The magic comes in on the input of DFF1. It is the XNOR of 2 or more outputs of the DFF. By picking the bits to XNOR carefully, you get an FSM that goes through a fairly random pattern, but with very simple hardware (note that these LFSRs can never go into the state with all 1’s, but reach all others). Two examples are given below.

![3-bit LFSR](Q1 Q2 Q3 D Q D Q D Q Clk)

Figure 1. 3-bit LFSR
Figure 2. 4-bit LFSR

First, draw the state diagram for these two circuits. It will show every possible state for the machine (8 for the 3-bit, 16 for the 4-bit), with arrows showing the next state they enter after that state.

Next, create a 9-bit LFSR in Quartus II and simulate it. You can find the list of bits to XNOR together in the table at the end of this lab (do NOT make up your own – most choices don’t work well, so the table shows the “best” connections to make).

Adder

In class we developed a circuit for adding binary numbers. Implement a 10-bit Adder circuit in Quartus and simulate it to show that it is working.

**ModelSim Tip:** now that you are working with multi-bit signals, it can be helpful in ModelSim to display signals as decimal or hex values. To do this, right-click on a multi-bit signal in ModelSim, and select “Radix”. Hex and Unsigned are the most useful choices IMHO.

CyberPlayer

We now have most of the components to implement a tunable cyber-player. First, let’s slow things down so you have a chance – run your entire Tug of War game off of the clock divider’s divided_clocks[15] (about 768Hz). To generate the computer’s button presses, add the LFSR output (a value from 0…511) to the value on SW[8:0] (a value from 0…511). If this value is greater than or equal to 512, count that as a button-press (BTW, how do you know if the result is >= 512? If it doesn’t seem trivial now, keep thinking about it…). You can speed up or slow down the system by simply playing with the user switches, to see how fast you can go. If the clock is too fast, feel free to adjust to a different divided_clock output (for the ENTIRE design). Note: be sure that EVERYTHING that is clocked in your design (except for the clock_divider circuit) uses the same clock. If you use any other clock then strange things can happen.

You will be graded 100 points on correctness, style, testing, testbenches, etc. Your bonus goal is developing the smallest circuit possible, measured in the same way as labs #5 & #6.

**Lab Demonstration/Turn-In Requirements**

A TA needs to "Check You Off" for each of the tasks listed below.

- Turn in your state diagrams derived from the 3-bit and 4-bit LFSRs.
- Turn in the Verilog for each of the modules developed in this lab, plus their testbenches.
- Demonstrate your complete system to the TA.
- Turn in a printout of the “Resource Utilization by Entity” page. Write on this the computed size for your design.
- Tell the TA how many hours (estimated) it took to complete this lab, including reading, planning, design, coding, debugging, testing, etc. Everything related to the lab (in total).

<table>
<thead>
<tr>
<th>n</th>
<th>XNOR from</th>
<th>n</th>
<th>XNOR from</th>
<th>n</th>
<th>XNOR from</th>
<th>n</th>
<th>XNOR from</th>
</tr>
</thead>
<tbody>
<tr>
<td>3</td>
<td>3,2</td>
<td>45</td>
<td>45,44,42,41</td>
<td>87</td>
<td>87,74</td>
<td>129</td>
<td>129,124</td>
</tr>
<tr>
<td>4</td>
<td>4,3</td>
<td>46</td>
<td>46,45,26,25</td>
<td>89</td>
<td>89,51</td>
<td>130</td>
<td>130,127</td>
</tr>
<tr>
<td>5</td>
<td>5,3</td>
<td>47</td>
<td>47,42</td>
<td>91</td>
<td>91,90,8,7</td>
<td>131</td>
<td>131,130,84,83</td>
</tr>
<tr>
<td>6</td>
<td>6,5</td>
<td>48</td>
<td>48,47,21,20</td>
<td>92</td>
<td>92,91,80,79</td>
<td>132</td>
<td>132,103</td>
</tr>
<tr>
<td>7</td>
<td>7,6</td>
<td>49</td>
<td>49,40</td>
<td>93</td>
<td>93,91</td>
<td>133</td>
<td>133,132,82,81</td>
</tr>
<tr>
<td>8</td>
<td>8,6,5,4</td>
<td>50</td>
<td>50,49,24,23</td>
<td>94</td>
<td>94,73</td>
<td>134</td>
<td>134,77</td>
</tr>
<tr>
<td>9</td>
<td>9,5</td>
<td>51</td>
<td>51,50,36,35</td>
<td>95</td>
<td>95,84</td>
<td>135</td>
<td>135,124</td>
</tr>
<tr>
<td>10</td>
<td>10,7</td>
<td>52</td>
<td>52,49</td>
<td>96</td>
<td>96,94,49,47</td>
<td>136</td>
<td>136,135,11,10</td>
</tr>
<tr>
<td>11</td>
<td>11,9</td>
<td>53</td>
<td>53,52,38,37</td>
<td>97</td>
<td>97,91</td>
<td>137</td>
<td>137,116</td>
</tr>
<tr>
<td>12</td>
<td>12,6,4,1</td>
<td>54</td>
<td>54,53,18,17</td>
<td>98</td>
<td>98,87</td>
<td>138</td>
<td>138,137,131,130</td>
</tr>
<tr>
<td>13</td>
<td>13,4,3,1</td>
<td>55</td>
<td>55,31</td>
<td>99</td>
<td>99,97,54,52</td>
<td>139</td>
<td>139,136,131,131</td>
</tr>
<tr>
<td>14</td>
<td>14,5,3,1</td>
<td>56</td>
<td>56,55,35,34</td>
<td>100</td>
<td>100,63</td>
<td>140</td>
<td>140,111</td>
</tr>
<tr>
<td>15</td>
<td>15,14</td>
<td>57</td>
<td>57,50</td>
<td>101</td>
<td>101,100,95,94</td>
<td>141</td>
<td>141,140,101,109</td>
</tr>
<tr>
<td>16</td>
<td>16,15,13,4</td>
<td>58</td>
<td>58,39</td>
<td>102</td>
<td>102,101,36,35</td>
<td>142</td>
<td>142,121</td>
</tr>
<tr>
<td>17</td>
<td>17,14</td>
<td>59</td>
<td>59,58,38,37</td>
<td>103</td>
<td>103,94</td>
<td>143</td>
<td>143,142,123,122</td>
</tr>
<tr>
<td>18</td>
<td>18,11</td>
<td>60</td>
<td>60,59</td>
<td>104</td>
<td>104,103,94,93</td>
<td>144</td>
<td>144,143,75,74</td>
</tr>
<tr>
<td>19</td>
<td>19,6,2,1</td>
<td>61</td>
<td>61,60,46,45</td>
<td>105</td>
<td>105,89</td>
<td>145</td>
<td>145,93</td>
</tr>
<tr>
<td>20</td>
<td>20,17</td>
<td>62</td>
<td>62,61,6,5</td>
<td>106</td>
<td>106,91</td>
<td>146</td>
<td>146,145,87,86</td>
</tr>
<tr>
<td>21</td>
<td>21,19</td>
<td>63</td>
<td>63,62</td>
<td>107</td>
<td>107,105,44,42</td>
<td>147</td>
<td>147,146,110,109</td>
</tr>
<tr>
<td>22</td>
<td>22,21</td>
<td>64</td>
<td>64,63,61,60</td>
<td>108</td>
<td>108,77</td>
<td>148</td>
<td>148,121</td>
</tr>
<tr>
<td>23</td>
<td>23,18</td>
<td>65</td>
<td>65,47</td>
<td>109</td>
<td>109,108,103,102</td>
<td>149</td>
<td>149,148,40,39</td>
</tr>
<tr>
<td>24</td>
<td>24,23,22,17</td>
<td>66</td>
<td>66,65,57,56</td>
<td>110</td>
<td>110,109,98,97</td>
<td>150</td>
<td>150,97</td>
</tr>
<tr>
<td>25</td>
<td>25,22</td>
<td>67</td>
<td>67,66,58,57</td>
<td>111</td>
<td>111,101,101,100</td>
<td>151</td>
<td>151,148</td>
</tr>
<tr>
<td>26</td>
<td>26,6,2,1</td>
<td>68</td>
<td>68,59</td>
<td>112</td>
<td>112,110,69,67</td>
<td>152</td>
<td>152,151,87,86</td>
</tr>
<tr>
<td>27</td>
<td>27,5,2,1</td>
<td>69</td>
<td>69,67,42,40</td>
<td>113</td>
<td>113,104,123,123</td>
<td></td>
<td></td>
</tr>
<tr>
<td>28</td>
<td>28,25</td>
<td>70</td>
<td>70,69,55,54</td>
<td>114</td>
<td>114,113,33,32</td>
<td>154</td>
<td>154,152,27,25</td>
</tr>
<tr>
<td>29</td>
<td>29,27</td>
<td>71</td>
<td>71,65</td>
<td>115</td>
<td>115,114,101,100</td>
<td>155</td>
<td>155,154,124,123</td>
</tr>
<tr>
<td>30</td>
<td>30,6,4,1</td>
<td>72</td>
<td>72,66,25,19</td>
<td>116</td>
<td>116,115,46,45</td>
<td>156</td>
<td>156,155,41,40</td>
</tr>
<tr>
<td>31</td>
<td>31,28</td>
<td>73</td>
<td>73,48</td>
<td>117</td>
<td>117,115,99,97</td>
<td>157</td>
<td>157,156,131,130</td>
</tr>
<tr>
<td>32</td>
<td>32,22,2,1</td>
<td>74</td>
<td>74,73,59,58</td>
<td>118</td>
<td>118,85</td>
<td>158</td>
<td>158,157,132,131</td>
</tr>
<tr>
<td>33</td>
<td>33,20</td>
<td>75</td>
<td>75,74,65,64</td>
<td>119</td>
<td>119,111,111,111</td>
<td>160</td>
<td>160,159,142,141</td>
</tr>
<tr>
<td>34</td>
<td>34,27,2,1</td>
<td>76</td>
<td>76,75,41,40</td>
<td>120</td>
<td>120,113,9,2</td>
<td>162</td>
<td>162,161,75,74</td>
</tr>
<tr>
<td>35</td>
<td>35,33</td>
<td>77</td>
<td>77,76,47,46</td>
<td>121</td>
<td>121,103,103,103</td>
<td>163</td>
<td>163,162,104,103</td>
</tr>
<tr>
<td>36</td>
<td>36,25</td>
<td>78</td>
<td>78,77,59,58</td>
<td>122</td>
<td>122,121,63,62</td>
<td>164</td>
<td>164,163,151,150</td>
</tr>
<tr>
<td>37</td>
<td>37,5,4,3,2,1</td>
<td>79</td>
<td>79,70</td>
<td>123</td>
<td>123,123,123,123</td>
<td>165</td>
<td>165,164,135,134</td>
</tr>
<tr>
<td>38</td>
<td>38,6,5,1</td>
<td>80</td>
<td>80,79,43,42</td>
<td>124</td>
<td>124,87</td>
<td>166</td>
<td>166,165,128,127</td>
</tr>
<tr>
<td>39</td>
<td>39,35</td>
<td>81</td>
<td>81,77</td>
<td>125</td>
<td>125,124,18,17</td>
<td>167</td>
<td>167,161</td>
</tr>
<tr>
<td>40</td>
<td>40,38,21,19</td>
<td>82</td>
<td>82,79,47,44</td>
<td>126</td>
<td>126,125,90,89</td>
<td>168</td>
<td>168,166,153,151</td>
</tr>
<tr>
<td>41</td>
<td>41,38</td>
<td>83</td>
<td>83,82,38,37</td>
<td>127</td>
<td>127,127</td>
<td>169</td>
<td>169,168,153,151</td>
</tr>
<tr>
<td>42</td>
<td>42,41,20,19</td>
<td>84</td>
<td>84,71</td>
<td>128</td>
<td>128,126,101,99</td>
<td>170</td>
<td>170,170,170,170</td>
</tr>
</tbody>
</table>

Figure 3. LFSR taps [XAPP 052 July 7, 1996 (Version 1.1), Peter Alfke, Xilinx Inc].