Instr:

ADD, XOR, OR, LOAD, STORE, BEQ, SLL, SRL, AND, XXR,

CPP - copy to r1

CYY - cppy to r2

Reg:

$r0 - Mem (load/store)

$r1,$r2 - Opearnd for R-type instruction

$r3 - Load from immediate

$r4-$r15 - general use

----------------------------------------------------------------

1. LDI 01000000 //r3=64 is the starting mem for input
2. CPP $r3 //r1 = 01000000 - 64
3. LDI 00000000
4. CYY $r3 //r2 = 0
5. ORR $r0 //r0 = 64  
   /// -----------------LOOP------------------
6. LOD $r4 //r4 = mem[64] LSW
7. LDI 00000001 //r3 = 1
8. CYY $r3 //r2 = r0
9. CPP $r0
10. ADD $r0 //r0 = 64 + 1
11. LOD $r5 //r5 = mem[65] MSB
12. LDI 00010110 //location of p4, p2, p1
13. CPP $r3 //r1 = 00010110
14. CYY $r4 //r4 = b4 b3 b2 p4 b1 p2 p1 p0
15. AND $r6 //r6 = 000(p4) 0(p2)(p1)0
16. CPP $r6 //r1 = 000(p4) 0(p2)(p1)0
17. LDI 00000001 //r3 = 1
18. CYY $r3 //r2 = 1
19. SRL $r6 //r6 = 0000 (p4)0(p2)(p1)
20. CPP $r6 //r1 = 0000 (p4)0(p2)(p1)
21. LDI 00001000
22. CYY $r3 //r2 = 00001000
23. AND $r1 //r1 = 0000(p4)000
24. LDI 00000001
25. CYY $r3 //r2 = 1
26. SRL $r7 //r7 = 0000 0(p4)00
27. CPP $r6 //r1 = 0000 (p4)0(p2)(p1)
28. CYY $r7 //r2 = 0000 0(p4)00
29. XOR $r1 //r1 = 0000 (P4)(p4)(p2)(p1)
30. LDI 00000111
31. CYY $r3
32. AND $r6 //r6 = 0000 0(p4)(p2)(p1)
33. LDI 00000001
34. CPP $r3
35. CYY $r5
36. AND $r1 //r1 = 0000 000(p8)
37. LDI 00000011 //r3 = 4
38. CYY $r3 //r2 = 3
39. SLL $r1 //r1 = 0000 (p8)0000
40. CYY $r6 //r2 = 0000 0(p4)(p2)(p1)
41. ORR $r14 //r14 = 0000 (p8)(p4)(p2)(p1)
42. LDI 11111110 //--------CAL P8-----------
43. CPP $r3 //r1
44. CYY $r5 //r2 = MSW
45. AND $r7 //r7 = b11 b10 b9 b8 b7 b6 b5 0
46. LDI 00000000
47. CPP $r3
48. CYY $r7 //r2 = r7 = b11 b10 b9 b8 b7 b6 b5 0
49. XXR $r1 //r1 = ^(b11,b10,b9,b8,b7,b6,b5,)
50. LDI 00000011 //li 3
51. CYY $r3 //r2 = 3
52. SLL $r15 //r15 = 0000 (p8)000
53. LDI 11100000 //--------CAL P4-----------
54. CPP $r3
55. CYY $r4 //r2 = LSW
56. AND $r6 //r6 = b4 b3 b2 0 0 0 0 0
57. LDI 11110000
58. CPP $r3
59. CYY $r5
60. AND $r7 //r7 = b11 b10 b9 b8 0 0 0 0
61. CPP $r6
62. CYY $r7
63. XXR $r6 //r6 = 0000000(p4)
64. LDI 00000010 // li 2
65. CPP $r6 // r1 = r6
66. CYY $r3 // r2 = 2
67. SLL $r6 // r6 = 0000 0(p4)00
68. CPP $r6 // r1 = 0000 0 (p4)00
69. CYY $r15 // r2 = 0000 (p8) 0 00
70. ORR $r15 // r15 = 0000 (p8)(p4)00
71. LDI 11001000 //--------CAL P2-----------
72. CPP $r3
73. CYY $r4 //r2 = LSW
74. AND $r6 //r6 = b4 b3 0 0 b1 0 0 0
75. LDI 11001100
76. CPP $r3
77. CYY $r5
78. AND $r7 //r7 = b11 b10 0 0 b7 b6 0 0
79. CPP $r6
80. CYY $r7
81. XXR $r6 //r6 = 0000000(p2)
82. LDI 00000001 // li 1
83. CPP $r6 // r1 = r6
84. CYY $r3 // r2 = 1
85. SLL $r6 // r6 = 0000 00(p2)0
86. CPP $r6 // r1 = 0000 00 (p2)0
87. CYY $r15 // r2 = 0000 p8 p4 0 0
88. ORR $r15 // r15 = 0000 p8 p4 p2 0
89. LDI 10101000 //--------CAL P1-----------
90. CPP $r3
91. CYY $r4 //r2 = LSW
92. AND $r6 //r6 = b4 0 b2 0 b1 0 0 0
93. LDI 10101010
94. CPP $r3
95. CYY $r5
96. AND $r7 //r7 = b11 0 b9 0 b7 0 b5 0
97. CPP $r6
98. CYY $r7
99. XXR $r6 //r6 = 0000000(p1)
100. CPP $r6 // r1 = 0 0 0 0 0 0 0 p1
101. CYY $r15 // r2 = 0000 p8 p4 p2 0
102. ORR $r15 // r15 = 0000 p8 p4 p2 p1
103. LDI 10011110 // address of 1 error - 158
104. CPP $r3
105. LDI 00000000
106. CYY $r3
107. ADD $r6 // r6 = addr of 1 error
108. CPP $r4
109. CYY $r5
110. XXR $r1 // r1 = ^LSW^MSW
111. LDI 00000000
112. CYY $r3 // r2 = 0
113. BNE $r6 // if ^LSW^MSW != 0 - go to 1error
114. CPP $r14 //r14 = parity bits
115. CYY $r15 //r15 = cal parity bits
116. XOR $r13 //error ptr = p8^c8 p4^c4 p2^c2 p1^c1
117. LDI 11010000 // add of 2 error - 208
118. CPP $r3
119. LDI 00000000
120. CYY $r3
121. ADD $r6 // r6 = addr of two error
122. LDI 00000000
123. CPP $r13 // r1 = err ptr
124. CYY $r3 // r2 = 0
125. BNE $r6 // if err\_ptr != 0 means 2 error:208
126. LDI 00011101 // -----No Error case/Write back-----
127. CPP $r3 // r1 = 29
128. CYY $r0 // r2 = mem
129. ADD $r0 // r0 = r0 + 29
130. STR $r4 // write back LSW
131. LDI 00000001
132. CPP $r3
133. CYY $r0
134. ADD $r0 //$r0 = $r0 + 1
135. STR $r5 //write back MSB
136. LDI 00011101 // li 29
137. CYY $r3
138. CPP $r0
139. SUB $r0 // r0 = r0 - 29
140. LDI 00000101 // starting of loop = 5
141. CPP $r3
142. LDI 00000000
143. CYY $r3
144. ADD $r6 // r6 = start address of LOOP
145. LDI 01011110 // li 94, prog end if read mem is 94
146. CPP $r3
147. CYY $r0
148. BNE $r6 // if not done branch to start
149. LDI 11011111 // end of program - 223
150. CPP $r3
151. LDI 00000000
152. CYY $r3
153. ADD $r6 //$r6 has end address
154. LDI 00000000
155. CPP $r3
156. LDI 00000001
157. CYY $r3
158. BNE $r6 // Unconditionally go to End program
159. LDI 10111011 // --------ONE ERROR------------
160. CPP $r3
161. LDI 00000000
162. CYY $r3
163. ADD $r6 // r6 = addr of MSB - 187
164. CPP $r14
165. CYY $r15
166. XOR $r13 // err\_ptr
167. LDI 00000111
168. CYY $r3 // r2 = 7
169. CPP $r13 // r1 = error ptr
170. BGT $r6 // if error ptr > 7 , goto MSB
171. LDI 00000001 // ---------EEROR IN LSW---------
172. CPP $r3
173. CYY $r13 // r2 = error\_ptr
174. SLL $r6 // r6 = 1'b1<<error\_ptr
175. CPP $r4 // r1 = LSW
176. CYY $r6 // r2 = 1'b1<<error\_ptr
177. XOR $r4 // r4 = LSW^(1'b1<<error\_ptr)
178. LDI 01111101 // address of Write-back - 125
179. CPP $r3
180. LDI 00000000
181. CYY $r3
182. ADD $r6 //r6 = 125 address of write-back
183. LDI 00000000
184. CPP $r3
185. LDI 00000001
186. CYY $r3
187. BNE $r6 // unconditionally go to write-back
188. LDI 00001000 // -------ERROR in MSB-------------
189. CYY $r3 // r2 = 8
190. CPP $r13 // r1 = error ptr
191. SUB $r13 // r13 = error ptr - 8
192. LDI 00000001
193. CPP $r3
194. CYY $r13 // r2 = error\_ptr
195. SLL $r6 // r6 = 1'b1<<error\_ptr
196. CPP $r5 // r1 = MSW
197. CYY $r6 // r2 = 1'b1<<error\_ptr
198. XOR $r5 // r4 = MSW^(1'b1<<error\_ptr)
199. LDI 01111101 // address of Write-back - 125
200. CPP $r3
201. LDI 00000000
202. CYY $r3
203. ADD $r6 //r6 = 125 address of write-back
204. LDI 00000000
205. CPP $r3
206. LDI 00000001
207. CYY $r3
208. BNE $r6 // unconditionally go to write-back
209. LDI 00000000 // ----------2 error----------------
210. CPP $r3
211. CYY $r4
212. AND $r4 //
213. AND $r5 // WRITE BACK ALL 0
214. LDI 01111101 // address of Write-back - 125
215. CPP $r3
216. LDI 00000000
217. CYY $r3
218. ADD $r6 //r6 = 125 address of write-back
219. LDI 00000000
220. CPP $r3
221. LDI 00000001
222. CYY $r3
223. BNE $r6 // unconditionally go to write-back
224. ----------------END OF PROGRAM---------------