Instr:

ADD, XOR, OR, LOD, STR, 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 00000000 //r3=0 is the starting mem for input
2. CPP $r3 //r1=0
3. CYY $r3
4. ORR $r0 //pass 0 to $r0  
   -------------------LOOP------------------------
5. LOD $r4 //r4 = mem[r0] , r0 = 0 at first
6. CPP $r0 // r1 = r0
7. LDI 00000001
8. CYY $r3 // r2 = 1
9. ADD $r0 // r0 = r0 + 1
10. LOD $r5 // r5 = mem[r0], r0 = 1 at first
11. LDI 11110000 // -----CAL P8-------
12. CPP $r3 // r1 = 11110000
13. CYY $r4 // r2 = LSW
14. AND $r6 // r6 = b8:b5
15. LDI 00000111
16. CPP $r3 //
17. CYY $r5 // r2 = MSW
18. AND $r7 // r7 = b11:b9
19. CPP $r6
20. CYY $r7
21. XXR $r15 // r15 = 0000 000(p8)
22. LDI 10001110 // -----CAL P4-------
23. CPP $r3 // r1 = 10001110
24. CYY $r4 // r2 = LSW
25. AND $r6 // r6 = b8,b4,b3,b2
26. LDI 00000111
27. CPP $r3 //
28. CYY $r5 // r2 = MSW
29. AND $r7 // r7 = b11:b9
30. CPP $r6
31. CYY $r7
32. XXR $r14 // r14 = 0000 000(p4)
33. LDI 00000100 // li 4
34. CPP $r14 // r1 = r14
35. CYY $r3 // r2 = 4
36. SLL $r14 // r14 = 000(p4) 0000
37. LDI 01101101 // -----CAL P2-------
38. CPP $r3 // r1 = 01101101
39. CYY $r4 // r2 = LSW
40. AND $r6 // r6 = b7,b6,b4,b3,b1
41. LDI 00000110
42. CPP $r3 //
43. CYY $r5 // r2 = MSW
44. AND $r7 // r7 = b11,b10
45. CPP $r6
46. CYY $r7
47. XXR $r13 // r13 = 0000 000(p2)
48. LDI 00000010 // li 2
49. CPP $r13 // r1 = r13
50. CYY $r3 // r2 = 2
51. SLL $r13 // r13 = 0000 0(p2)00
52. CPP $r13
53. CYY $r14
54. ORR $r14 r14 = 000(p4) 0(p2)00
55. LDI 01011011 // -----CAL P1-------
56. CPP $r3 // r1 = 01011011
57. CYY $r4 // r2 = LSW
58. AND $r6 // r6 = b7,b5,b4,b2,b1
59. LDI 00000101
60. CPP $r3 //
61. CYY $r5 // r2 = MSW
62. AND $r7 // r7 = b11,b9
63. CPP $r6
64. CYY $r7
65. XXR $r13 // r13 = 0000 000(p1)
66. LDI 00000001 // li 1
67. CPP $r13 // r1 = r13
68. CYY $r3 // r2 = 2
69. SLL $r13 // r13 = 0000 00(p1)0
70. CPP $r13
71. CYY $r14
72. ORR $r14 // r14 = 000(p4) 0(p2)(p1)0
73. CPP $r4 --------CAL P16---------
74. CYY $r5
75. XXR $r6 // r6 LSbit has ^(b11:1)
76. CPP $r15
77. CYY $r14
78. XXR $r7 // r7 LSbit has ^(p8,p4,p2,p1)
79. CPP $r6
80. CYY $r7
81. XXR $r6 // r6 LSbit has ^(b11:1,p8,p4,p2,p1)
82. CPP $r6
83. CYY $r14
84. ADD $r14 // r14 = 000(p4) 0(p2)(p1)(p16)
85. LDI 00000101 // li 5
86. CYY $r3 // r2 = 5
87. CPP $r5 // r1 = MSW
88. SLL $r6 // r6 = b11 b10 b9 0 0 0 0 0
89. LDI 00000011 // li 3
90. CYY $r3 // r2 = 3
91. CPP $r4 // r1 = LSW
92. SRL $r7 // r7 = 0 0 0 b8 b7 b6 b5 b4
93. LDI 00011110
94. CPP $r3 // r1 = 00011110
95. CYY $r7
96. AND $r7 // r7 = 0 0 0 b8 b7 b6 b5 0
97. CPP $r15 // r1 = 000 0 0 0 0 (p8)
98. CYY $r7 // r7 = 000 b8 b7 b6 b5 0
99. XOR $r7 // r7 = 000 b8 b7 b6 b5 p8
100. CPP $r7
101. CYY $r6
102. XOR $r15 // r15 = b11 b10 b9 b8 b7 b6 b5 p8
103. LDI 00000100 // li 4
104. CYY $r3
105. CPP $r4
106. SLL $r6 // r6 = b4 b3 b2 b1 0 0 0 0
107. LDI 00010000
108. CPP $r3
109. CYY $r6
110. AND $r7 // r7 = 0 0 0 b1 0 0 0 0
111. LDI 00000001
112. CYY $r3
113. CPP $r7
114. SRL $r7 // r7 = 0 0 0 0 b1 0 0 0
115. CPP $r6
116. CYY $r7
117. XOR $r6 // r6 = b4 b3 b2 b1 b1 0 0 0
118. LDI 11101111
119. CPP $r3 // r1 = 1 1 1 0 1 1 1 1
120. CYY $r6 // r2 = b4 b3 b2 b1 b1 0 0 0
121. AND $r6 // r6 = b4 b3 b2 0 b1 0 0 0
122. CPP $r14 // r1 = 0 0 0 (p4) 0 (p2)(p1)(p16)
123. CYY $r6 // r2 = b4 b3 b2 0 b1 0 0 0
124. ORR $r14 // r14 = b4 b3 b2 p4 b1 p2 p1 p16
125. LDI 00011101 // li 29
126. CPP $r0 // r1 = 1
127. CYY $r3 // r3 = 29
128. ADD $r0 // r0 = 30
129. STR $r14 // Mem[30] = b4 b3 b2 p4 b1 p2 p1 p16
130. LDI 00000001 // LI 1
131. CPP $r0
132. CYY $r3
133. ADD $r0 //r0 = r0 + 1 = 31
134. STR $15 //mem[31] = b11 b10 b9 b8 b7 b6 b5 p8
135. LDI 00000100 // Start address + 5
136. CPP $r3
137. LDI 00000000
138. CYY $r3
139. ADD $r6 // r6 = 5
140. LDI 00011101 // li 29
141. CYY $r3
142. CPP $r0
143. SUB $r0 // r0 = r0 - 29
144. LDI 00011110 // li 30, prog end if read mem is 30
145. CPP $r3
146. CYY $r0
147. BNE $r6 //if r0!=30 , go to loop, if r0=30 end
148. 111111111 ----------END OF PROG 1-------------