```
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 10100000 //mem[160] = starting reading address
2. CPP $r3
3. LDI 00000000
4. CYY $r3
5. ORR $r0
6. LOD $r1 // r1 = x x x x x _ 000 - 5 bits x pattern
7. LDI 00000011
8. CYY $r3
9. SRL $r12 //r12 = 000_xxxxx - bits x pattern
10. LDI 10000000
11. CPP $r3 //mem[128] start of reading
12. LDI 00000000
13. CYY $r3
14. ORR $r0 //r0 = 128
15. LDI 00000000
16. CPP $r3
17. CYY $r3
18. ADD $r13
19. ADD $r14
20. ADD $r14

    ----------LOOP----------

21. LOD $r4 //r4 = mem[128]
22. LDI 00011111 // first 5 bit
23. CPP $r3
24. CYY $r4
25. AND $r6 //r6 = {0,0,0, r4[4:0]}
26. LDI 00111110
27. CPP $r3
28. AND $r1
29. LDI 00000001
30. CYY $r3
31. SRL $r7 //r7= {0,0,0,r4[5:1]}
32. LDI 01111100

```
CPP $r3
CYY $r4
AND $r1
LDI 00000010
CYY $r3
SRL $r8                    //r8={0,0,0,r4[6:2]}
LDI 11111000
CPP $r3
CYY $r4
AND $r1
LDI 00000011
CYY $r3
SRL $r9                    //r9={0,0,0,r4[7:3]}
    LDI 00000000
    CPP $r3
    AND $r11               // r11 is flag for occur in a byte
CPP $r6
CYY $r12
EQL $r1                    // if r6 == pattern r1 = 1
CYY $r13                   // Occurrence counter
ADD $r13                   // r13 = r13 + r1
    CYY  $r11
    ORR $r11               // r11 ever = 1 will stay 1
CPP $r7
CYY $r12
EQL $r1                    // if r7 == pattern r1 = 1
CYY $r13                   // Occurrence counter
ADD $13                    // r13 = r13 + r1
    CYY  $r11
    ORR $r11               // r11 ever = 1 will stay 1
CPP $r8
CYY $r12
EQL $r1                    // if r8 == pattern r1 = 1
CYY $r13                   // Occurrence counter
ADD $13                    // r13 = r13 + r1
    CYY  $r11
    ORR $r11               // r11 ever = 1 will stay 1
CPP $r9
CYY $r12
EQL $r1                    // if r9 == pattern r1 = 1
CYY $r13                   // Occurrence counter
ADD $r13                   // r13 = r13 + r1
    CYY  $r11
    ORR $r11               // r11 ever = 1 will stay 1
     CPP $r14
```

```
78.         CYY $r11
79.         ADD $r14
80. LDI 00010100                    // start loop - 20
81. CPP $r3
82. LDI 00000000
83. CYY $r3
84. ADD $r6                         //r6 = 14
85. LDI 00000001
86. CPP $r3
87. CYY $r0
88. ADD $r0                         //r0++
89. LDI 10100000                    // 160 end of read
90. CPP $r3
91. CYY $r0
92. BNE 1001                                // r0 != 150 go to loop
93. LDI 11000000                    // else str mem[192]
94. CPP $r3
95. LDI 00000000
96. CYY $r3
97. ADD $r0
98. STR $r13
99. LDI 00000001
100.    CPP $r3
101.    CYY $r0
102.    ADD $r0
103.    STR $r14
    ------------------------------------LAST PART--------------------------
104.    LDI 10000000
105.    CPP $r3                                 //mem[128] start of reading
106.    LDI 00000000
107.    CYY $r3
108.    ORR $r0                         //r0 = 128
    ---------------------------------LOOP---------------------------------
109.    LOD $r5                         //r5 = mem[128]
110.    LDI 00000001
111.    CPP $r3
112.    CYY $r0
113.    ADD $r0                         //r0= r0 + 1
114.    LOD $r4                         //r4=mem[129]
115.    LDI 11110000
116.    CPP $r3
117.    CYY $r4
118.    AND $r6                         //r6 = a7 a6 a5 a4 0 0 0 0
119.    LDI 00000100
120.    CYY $r3
```

```
121.  CPP $r6
122.  SRL $r6                    // r6 = 0 0 0 0 a7 a6 a5 a4
123.  LDI 00001111
124.  CPP $r3
125.  CYY $r5
126.  AND $r7          //r7 = 0 0 0 0 b3 b2 b1 b0
127.  LDI 00000100
128.  CYY $r3
129.  CPP $r7
130.  SLL $r1                    //r7 = b3 b2 b1 b0 0 0 0 0
131.  CYY $r6
132.  ORR $r10          //r10 = b3 b2 b1 b0 a7 a6 a5 a4
133.  LDI 00011111              // first 5 bit
134.  CPP $r3
135.  CYY $r10
136.  AND $r6          //r6 = {0,0,0, r4[4:0]}
137.  LDI 00111110
138.  CPP $r3
139.  AND $r1
140.  LDI 00000001
141.  CYY $r3
142.  SRL $r7                    //r7= {0,0,0,r4[5:1]}
143.  LDI 01111100
144.  CPP $r3
145.  CYY $r10
146.  AND $r1
147.  LDI 00000010
148.  CYY $r3
149.  SRL $r8                    //r8={0,0,0,r4[6:2]}
150.  LDI 11111000
151.  CPP $r3
152.  CYY $r10
153.  AND $r1
154.  LDI 00000011
155.  CYY $r3
156.  SRL $r9                    //r9={0,0,0,r4[7:3]}
157.  CPP $r6
158.  CYY $r12
159.  EQL $r1                          // if r6 == pattern r1 = 1
160.  CYY $r15                  // Occurrence counter
161.  ADD $r15                  // r13 = r13 + r1
162.  CPP $r7
163.  CYY $r12
164.  EQL $r1                          // if r7 == pattern r1 = 1
165.  CYY $r15                  // Occurrence counter
```

```
166.  ADD $r15              // r13 = r13 + r1
167.  CPP $r8
168.  CYY $r12
169.  EQL $r1                    // if r8 == pattern r1 = 1
170.  CYY $r15              // Occurrence counter
171.  ADD $r15
172.  CPP $r9
173.  CYY $r12
174.  EQL $r1                    // if r9 == pattern r1 = 1
175.  CYY $r15              // Occurrence counter
176.  ADD $r15
177.  LDI 01101100               // start loop - 108
178.  CPP $r3
179.  LDI 00000000
180.  CYY $r3
181.  ADD $r6               //r6 = 108
182.  LDI 10011111               // 159 end of read
183.  CPP $r3
184.  CYY $r0
185.  BNE 1011                   // r0 != 159 go to loop
186.  CPP $r13
187.  CYY $r15
188.  ADD $r15              //r15 =r13+r15
189.  LDI 11000010               // 194
190.  CPP $r3
191.  LDI 00000000
192.  CYY $r3
193.  ORR $r0                    //r0 = 194
194.  STR $r15
```