UNIVERSIDADE FEDERAL DE OURO PRETO
CURSO DE GRADUAÇÃO EM CIÊNCIA DA COMPUTAÇÃO

CRISTIAN ANSELMO DE SOUZA PINTO - 21.2.4005
CÉSAR GABRIEL DE PAULA BATISTA - 21.2.4176
IVES HENRIQUE SENIBALDE DE OLIVEIRA- 21.2.4184
JULIANA APARECIDA BORGES - 21.2.4156
LUCAS DE OLIVEIRA BARBOZA- 21.2.4173
MARIA CLARA MIRANDA DE SÁ- 21.2.4187

ATIVIDADE ABERTA 02 DE ELETRÔNICA PARA COMPUTAÇÃO
BCC-265

OURO PRETO
2022

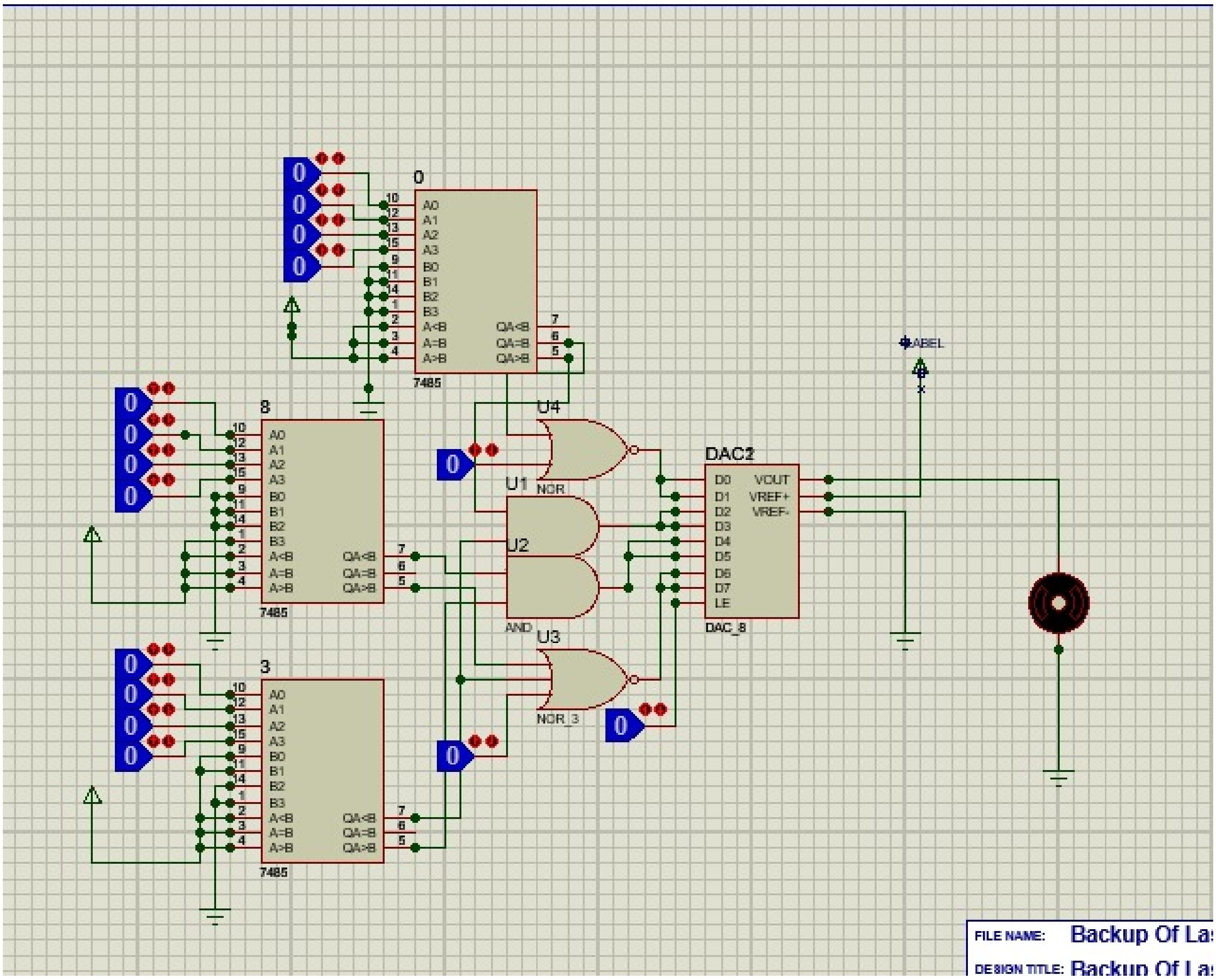
Nessa atividade aberta tivemos que criar um circuito no Proteus, onde teria que supor a necessidade de atuar sobre 4 motores DC. Apenas um motor deveria ser ativado por vez onde a sua seleção seria feita por intermédio de uma palavra de seleção de dois bits (bits S1 e S0). Além da seleção do motor, o módulo também receberia uma palavra indicativa de velocidade composta por 4 bits (bits V3, V2, V1 e V0). Externamente, cada motor teria a sua ligação feita através de um conversor digital-analógico (DAC).

Para controlarmos a velocidade do motor, o módulo de controle, a partir da palavra “V” (relativa à velocidade) faria o seguinte tratamento:

```
se V==0 então “Velocidade=0”
else se V < 3 então “Velocidade 1”
else se V < 8 então “Velocidade 2”
else “Velocidade 3”.
```

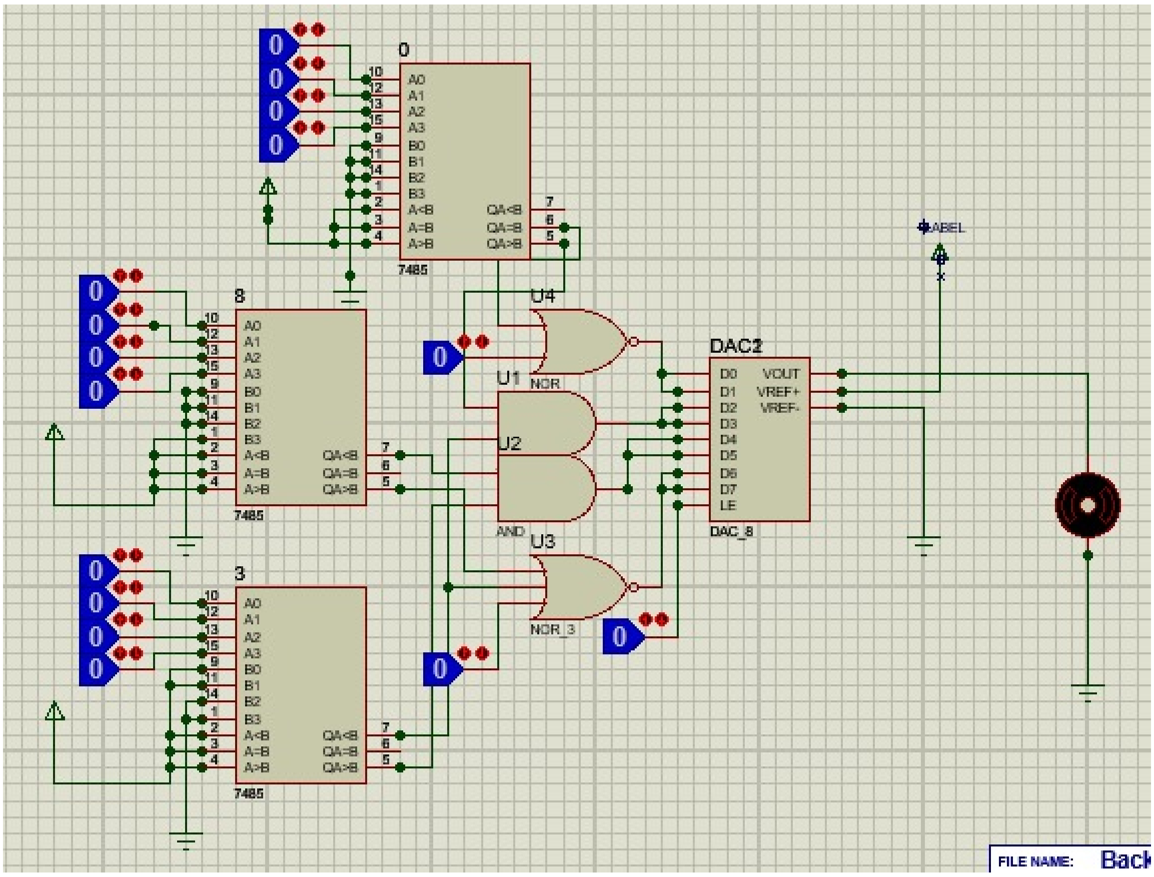
Os bits resultantes do tratamento de “V” seriam ligados, externamente, a um DAC – cuja saída atuará no motor.

Para que conseguíssemos as referidas velocidades, as entradas do DAC deveriam ser ligadas, respectivamente, às seguintes palavras: “00001111” (Velocidade 1), “00110011” (Velocidade 2) e “11000011” (Velocidade 3). A “Velocidade=0” indica uma saída formada pela palavra “00000000”.



FILE NAME: Backup Of La

DESIGN TITLE: Backup Of La



VERILOG

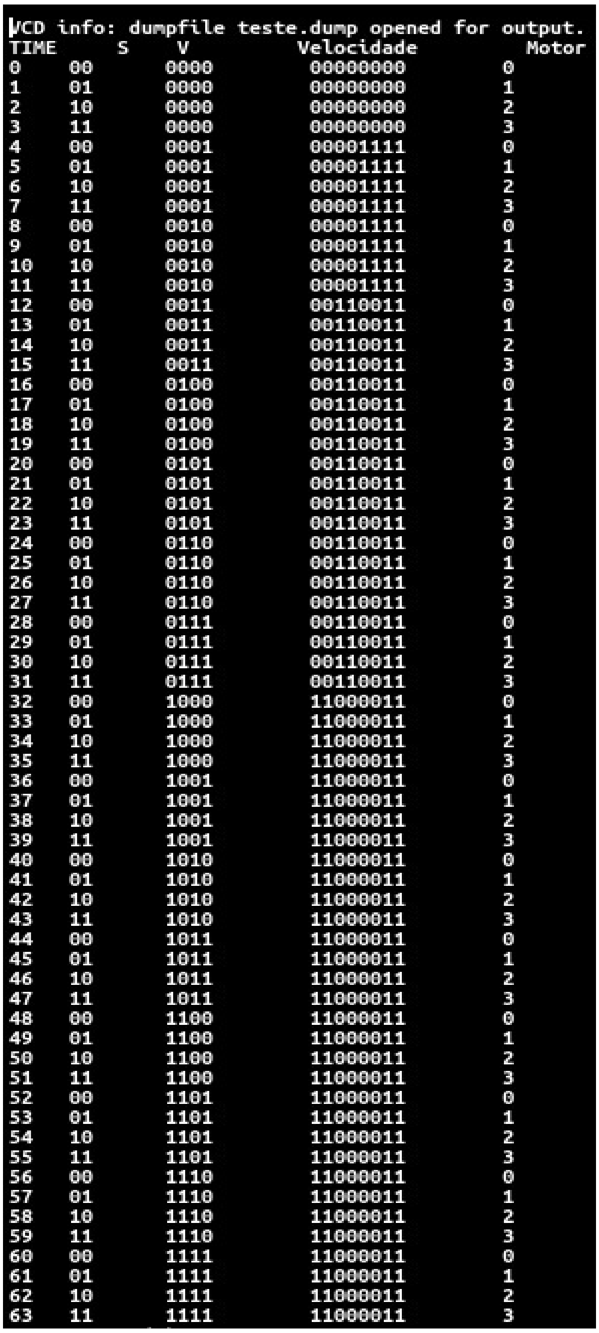
Código:

S1 e S0 controlam qual dos 4 motores vai ser ligado
00 - motor 0
01 - motor 1
10 - motor 2
11 - motor 3

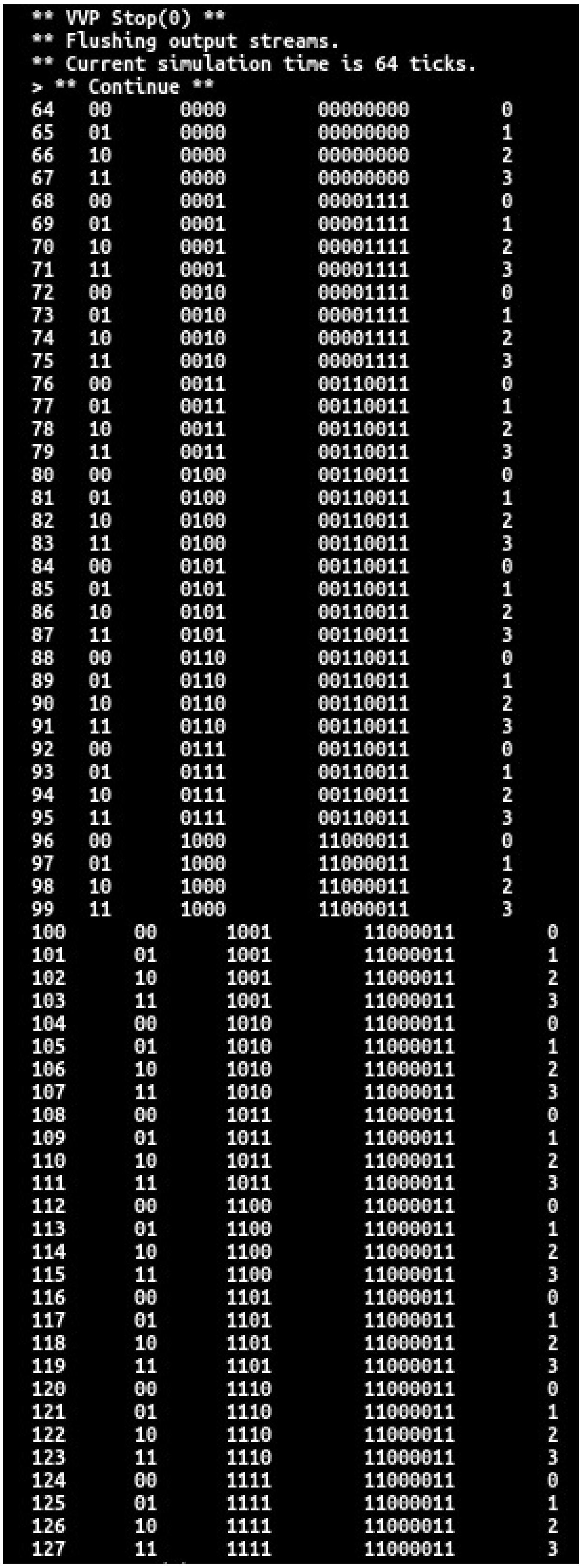
V -> palavra de 4 bits controla a velocidade
(V3, V2, V1, V0)
0000-velocidade 0
0001-velocidade 1
0010-velocidade 1
0011-velocidade 2
0100-velocidade 2
0101-velocidade 2
0110-velocidade 2
0111-velocidade 2
1000-velocidade 2
1001-velocidade 3
1010-velocidade 3
1011-velocidade 3
1100-velocidade 3
1101-velocidade 3

```
1110-velocidade 3
1111-velocidade 3

*/
module Controle(S, Velocidade, V, Motor);

    input [3:0] V;
    input [1:0] S;
    output reg [7:0] Velocidade;
    output reg [1:0] Motor;

    always @(*)
        begin
            if (S == 0)
                begin
                    Motor = 2'b00;
                    if (V == 0) Velocidade = 0;
                    else if (V < 3) Velocidade = 8'b00001111;
                    else if (V < 8) Velocidade = 8'b00110011;
                    else Velocidade = 8'b11000011;
                end
            if (S == 1)
                begin
                    Motor = 2'b01;
                    if (V == 0) Velocidade = 0;
                    else if (V < 3) Velocidade = 8'b00001111;
                    else if (V < 8) Velocidade = 8'b00110011;
                    else Velocidade = 8'b11000011;
                end
            if (S == 2)
                begin
                    Motor = 2'b10;
                    if (V == 0) Velocidade = 0;
                    else if (V < 3) Velocidade = 8'b00001111;
                    else if (V < 8) Velocidade = 8'b00110011;
                    else Velocidade = 8'b11000011;
                end
            if (S == 3)
                begin
                    Motor = 2'b11;
                    if (V == 0) Velocidade = 0;
                    else if (V < 3) Velocidade = 8'b00001111;
                    else if (V < 8) Velocidade = 8'b00110011;
                    else Velocidade = 8'b11000011;
                end
        end

endmodule //Controle

module top();

  reg [3:0] a;
  reg [1:0] b;
  wire [7:0] x;
  wire [1:0] y;

  initial
    begin
      a = 4'b0000;
      b = 2'b00;
    end

  always begin
    begin
      #64 $stop;
    end
  end

  always begin
    #1 {a,b} = {a,b} + 1;
  end

  initial
    begin
      $dumpfile("teste.dump");
        $dumpvars(0,a,b,x,y);
        $dumpon;
        $display("TIME \t S \t  V \t    Velocidade \t       Motor");
        $monitor("%0d \t %b \t %b \t     %b \t\t %d ",$time,b,a,x,y);
    end

  Controle integ(.S(b), .Velocidade(x), .V(a), .Motor(y));

endmodule
```

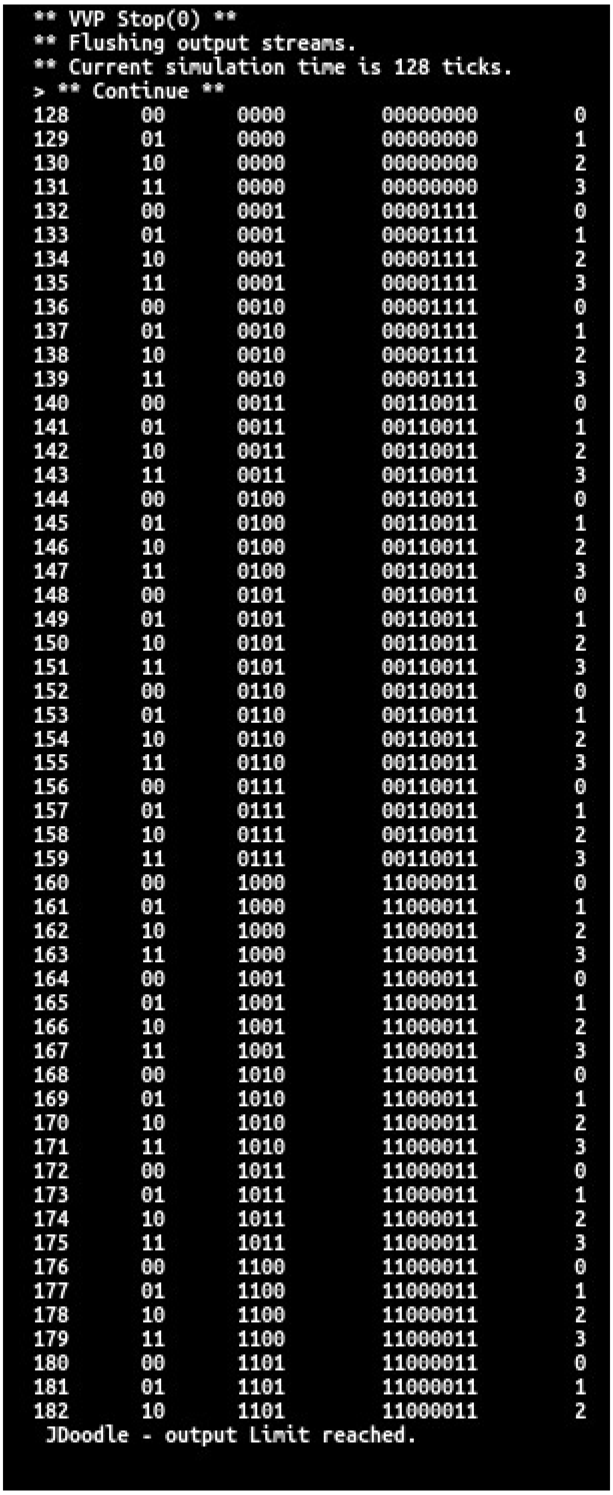
Compilação:



```
VCD info: dumpfile teste.dump opened for output.
TIME      S     V           Velocidade         Motor
0     00       0000          00000000         0
1     01       0000          00000000         1
2     10       0000          00000000         2
3     11       0000          00000000         3
4     00       0001          00001111         0
5     01       0001          00001111         1
6     10       0001          00001111         2
7     11       0001          00001111         3
8     00       0010          00001111         0
9     01       0010          00001111         1
10    10       0010          00001111         2
11    11       0010          00001111         3
12    00       0011          00110011         0
13    01       0011          00110011         1
14    10       0011          00110011         2
15    11       0011          00110011         3
16    00       0100          00110011         0
17    01       0100          00110011         1
18    10       0100          00110011         2
19    11       0100          00110011         3
20    00       0101          00110011         0
21    01       0101          00110011         1
22    10       0101          00110011         2
23    11       0101          00110011         3
24    00       0110          00110011         0
25    01       0110          00110011         1
26    10       0110          00110011         2
27    11       0110          00110011         3
28    00       0111          00110011         0
29    01       0111          00110011         1
30    10       0111          00110011         2
31    11       0111          00110011         3
32    00       1000          11000011         0
33    01       1000          11000011         1
34    10       1000          11000011         2
35    11       1000          11000011         3
36    00       1001          11000011         0
37    01       1001          11000011         1
38    10       1001          11000011         2
39    11       1001          11000011         3
40    00       1010          11000011         0
41    01       1010          11000011         1
42    10       1010          11000011         2
43    11       1010          11000011         3
44    00       1011          11000011         0
45    01       1011          11000011         1
46    10       1011          11000011         2
47    11       1011          11000011         3
48    00       1100          11000011         0
49    01       1100          11000011         1
50    10       1100          11000011         2
51    11       1100          11000011         3
52    00       1101          11000011         0
53    01       1101          11000011         1
54    10       1101          11000011         2
55    11       1101          11000011         3
56    00       1110          11000011         0
57    01       1110          11000011         1
58    10       1110          11000011         2
59    11       1110          11000011         3
60    00       1111          11000011         0
61    01       1111          11000011         1
62    10       1111          11000011         2
63    11       1111          11000011         3
```



```
** VVP Stop(0) **
** Flushing output streams.
** Current simulation time is 64 ticks.
> ** Continue **
64   00      0000        00000000        0
65   01      0000        00000000        1
66   10      0000        00000000        2
67   11      0000        00000000        3
68   00      0001        00001111        0
69   01      0001        00001111        1
70   10      0001        00001111        2
71   11      0001        00001111        3
72   00      0010        00001111        0
73   01      0010        00001111        1
74   10      0010        00001111        2
75   11      0010        00001111        3
76   00      0011        00110011        0
77   01      0011        00110011        1
78   10      0011        00110011        2
79   11      0011        00110011        3
80   00      0100        00110011        0
81   01      0100        00110011        1
82   10      0100        00110011        2
83   11      0100        00110011        3
84   00      0101        00110011        0
85   01      0101        00110011        1
86   10      0101        00110011        2
87   11      0101        00110011        3
88   00      0110        00110011        0
89   01      0110        00110011        1
90   10      0110        00110011        2
91   11      0110        00110011        3
92   00      0111        00110011        0
93   01      0111        00110011        1
94   10      0111        00110011        2
95   11      0111        00110011        3
96   00      1000        11000011        0
97   01      1000        11000011        1
98   10      1000        11000011        2
99   11      1000        11000011        3
100       00      1001        11000011        0
101       01      1001        11000011        1
102       10      1001        11000011        2
103       11      1001        11000011        3
104       00      1010        11000011        0
105       01      1010        11000011        1
106       10      1010        11000011        2
107       11      1010        11000011        3
108       00      1011        11000011        0
109       01      1011        11000011        1
110       10      1011        11000011        2
111       11      1011        11000011        3
112       00      1100        11000011        0
113       01      1100        11000011        1
114       10      1100        11000011        2
115       11      1100        11000011        3
116       00      1101        11000011        0
117       01      1101        11000011        1
118       10      1101        11000011        2
119       11      1101        11000011        3
120       00      1110        11000011        0
121       01      1110        11000011        1
122       10      1110        11000011        2
123       11      1110        11000011        3
124       00      1111        11000011        0
125       01      1111        11000011        1
126       10      1111        11000011        2
127       11      1111        11000011        3
```



```
** VVP Stop(0) **
** Flushing output streams.
** Current simulation time is 128 ticks.
> ** Continue **
128      00      0000        00000000        0
129      01      0000        00000000        1
130      10      0000        00000000        2
131      11      0000        00000000        3
132      00      0001        00001111        0
133      01      0001        00001111        1
134      10      0001        00001111        2
135      11      0001        00001111        3
136      00      0010        00001111        0
137      01      0010        00001111        1
138      10      0010        00001111        2
139      11      0010        00001111        3
140      00      0011        00110011        0
141      01      0011        00110011        1
142      10      0011        00110011        2
143      11      0011        00110011        3
144      00      0100        00110011        0
145      01      0100        00110011        1
146      10      0100        00110011        2
147      11      0100        00110011        3
148      00      0101        00110011        0
149      01      0101        00110011        1
150      10      0101        00110011        2
151      11      0101        00110011        3
152      00      0110        00110011        0
153      01      0110        00110011        1
154      10      0110        00110011        2
155      11      0110        00110011        3
156      00      0111        00110011        0
157      01      0111        00110011        1
158      10      0111        00110011        2
159      11      0111        00110011        3
160      00      1000        11000011        0
161      01      1000        11000011        1
162      10      1000        11000011        2
163      11      1000        11000011        3
164      00      1001        11000011        0
165      01      1001        11000011        1
166      10      1001        11000011        2
167      11      1001        11000011        3
168      00      1010        11000011        0
169      01      1010        11000011        1
170      10      1010        11000011        2
171      11      1010        11000011        3
172      00      1011        11000011        0
173      01      1011        11000011        1
174      10      1011        11000011        2
175      11      1011        11000011        3
176      00      1100        11000011        0
177      01      1100        11000011        1
178      10      1100        11000011        2
179      11      1100        11000011        3
180      00      1101        11000011        0
181      01      1101        11000011        1
182      10      1101        11000011        2
 JDoodle - output Limit reached.
```