

MORS KODU DÖNÜŞTÜRÜCÜ

Mustafa Uğur GÜL 203405031

Gökay HELVACI 203405033

Aa	• -	Jj	• - -	Ss	... -
Bb	- • -	Kk	- - .	Tt	- -
Cc	- - • -	Ll	- - ..	Uu	.. -
Dd	- - - • -	Mm	- - -	Vv	... -
Ee	- - - - • -	Nn	- - -	Ww	- - -
Ff	- - - - - • -	Oo	- - - -	Xx	- - -
Gg	- - - - - - • -	Pp	- - - - -	Yy	- - -
Hh	- - - - - - - • -	Qq	- - - - - -	Zz	- - - -
Ii	- - - - - - - - • -	Rr	- - - - - - -		

FLOOR PLANNER VE CST FILE



```
1 //Copyright (C)2014-2023 Gowin Semiconductor Corporation.
2 //All rights reserved.
3 //File Title: Physical Constraints file
4 //GOWIN Version: 1.9.8.11
5 //Part Number: GW1NR-LV9QN88PC6/I5
6 //Device: GW1NR-9
7 //Device Version: C
8 //Created Time: Sun 05 28 10:39:53 2023
9
10 IO_LOC "leds[5]" 16;
11 IO_PORT "leds[5]" PULL_MODE=UP DRIVE=8 BANK_VCCIO=1.8;
12 IO_LOC "leds[4]" 15;
13 IO_PORT "leds[4]" PULL_MODE=UP DRIVE=8 BANK_VCCIO=1.8;
14 IO_LOC "leds[3]" 14;
15 IO_PORT "leds[3]" PULL_MODE=UP DRIVE=8 BANK_VCCIO=1.8;
16 IO_LOC "leds[2]" 13;
17 IO_PORT "leds[2]" PULL_MODE=UP DRIVE=8 BANK_VCCIO=1.8;
18 IO_LOC "leds[1]" 11;
19 IO_PORT "leds[1]" PULL_MODE=UP DRIVE=8 BANK_VCCIO=1.8;
20 IO_LOC "leds[0]" 10;
21 IO_PORT "leds[0]" PULL_MODE=UP DRIVE=8 BANK_VCCIO=1.8;
22 IO_LOC "rst" 4;
23 IO_PORT "rst" PULL_MODE=UP BANK_VCCIO=1.8;
24 IO_LOC "clk" 52;
25 IO_PORT "clk" IO_TYPE=LVC MOS33 PULL_MODE=UP BANK_VCCIO=3.3;
```

```

1 module StringSplitter(
2     input [63:0] str,
3     output reg [4:0][7:0] letters
4 );
5     integer i;
6
7     always @(*) begin
8         for (i = 0; i < 5; i = i + 1) begin
9             letters[i] = str[(i*8)+:8];
10        end
11    end
12
13 endmodule
14
15 module subModule(
16     input clk,
17     input rst,
18     output reg [5:0] leds
19 );
20
21 reg [63:0] str; // String'i temsil etmek için 64-bit değişken
22 wire [4:0][7:0] letters_wire; // Harfleri temsil etmek için 5 adet 8-bit wire sinyali
23 reg [4:0][7:0] letters; // Harfleri temsil etmek için 5 adet 8-bit reg sinyali
24 integer i;
25 reg [1:0] state;
26 reg run;
27

```

- ▶ String Splitter 64 bitlik bir stringi alıp letters'a harfler ters olacak şekilde (ilk harf 64 bitte son harf sıfırıncı bitte olacak şekilde) atar.

- Submodule ise clock, reset inputlarını ve reg outputunu içerir.

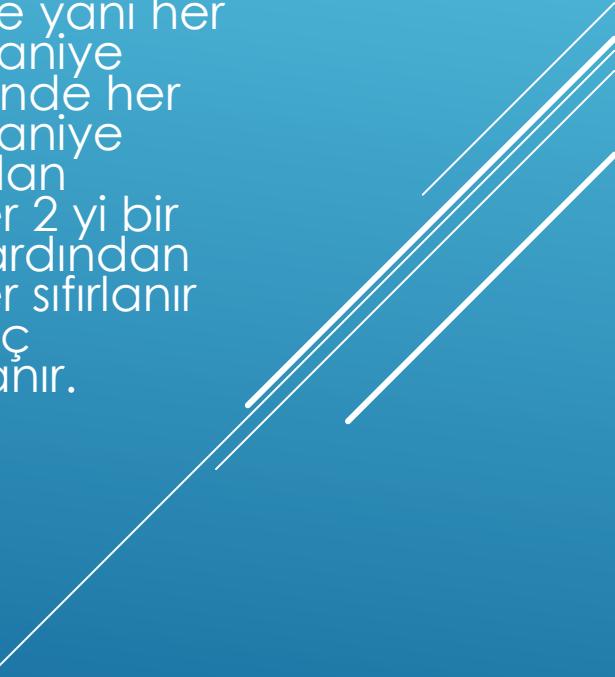
```

30 initial begin
31     str = "selam"; // ASCII değerlerini kullanarak "selam" stringini atayın
32     i=4;
33     run=0;
34     leds <= 6'b111111;
35 end
36
37
38
39
40 StringSplitter splitter(
41     .str(str),
42     .letters(letters_wire)
43 );
44
45
46 reg [23:0] counter;
47 reg[5:0] counter2;
48
49
50
51 always @ (posedge clk or negedge rst) begin
52
53     if(!rst) begin
54         counter <= 24'd0; // butondan reset çağrısı geldiğinde counterlar sıfırlanır
55         counter2 <= 6'd0;
56     end
57     else if (counter2 > 6'd12)
58         counter2 <=6'd0; // 17. saniyeden sonra tekrar sıfırıncı saniyeye geçilir
59     else if( counter < 24'd1349_9999)
60         counter <= counter + 1;
61     else begin
62         counter2 <= counter2 + 1;
63         counter <= 24'd0;
64     end
65
66
67
68 end
69
70
71 always @ (posedge clk or posedge rst) begin

```

- ▶ Initial begin kısmında değer atama yapılır. StringSplitter harfleri ters olarak atıldığından iyi son harf olan 4 ten başlattık.

Eğer counter 1349999 sayısını geçerse yani her yarımsaniye geçtiğinde her yarımsaniye sayısı olan counter 2 yi bir arttırır ardından counter sıfırlanır ve süreç tekrarlanır.



```
76    if (!rst) begin
77        leds <= 6'b111111;
78        run=0;
79        i = 4;
80
81    end else begin
82        letters <= letters_wire; // letters_wire sinyalini letters reg sinyaline atayın
83        leds[5:0] <= 6'b111111;
84
85        case (letters[i])
86            "a": begin
87
88
89                if(!run && counter2 == 6'd0)
90                    run=1;
91            if(run) begin
92                if (counter2 < 6'd1)
93                    leds[5:0] <= 6'b000000;
94                else if ( counter2 < 6'd2)
95                    leds[5:0] <= 6'b111111;
96                else if ( counter2 < 6'd5)
97                    leds[5:0] <= 6'b000000;
98                else if ( counter2 < 6'd6)
99                    leds[5:0] <= 6'b111111;
100            else begin
101                leds[5:0] <= leds;
102                i = i - 1;
103                run=0;
104            end
105        end
106    end
107
```

Decimal bitler
1 olduğunda
ledler söner 0
olduğunda ise
yanar. Eğer bir
led 0.5 saniye
yanıyorsa kısa,
1.5 saniye
yanıyorsa uzun
kabul edilir ve
bu şekilde latin
alfabesi mors
koduna çevirilir
ardından bir
sonraki harfi
okumak için i
bir azaltılır.

```
107     "b": begin
108
109         if(!run && counter2 == 6'd0)
110             run=1;
111         if(run) begin
112             if (counter2 < 6'd3)
113                 leds[5:0] <= 6'b000000;
114             else if ( counter2 < 6'd4)
115                 leds[5:0] <= 6'b111111;
116             else if ( counter2 < 6'd5)
117                 leds[5:0] <= 6'b000000;
118             else if ( counter2 < 6'd6)
119                 leds[5:0] <= 6'b111111;
120             else if ( counter2 < 6'd7)
121                 leds[5:0] <= 6'b000000;
122             else if ( counter2 < 6'd8)
123                 leds[5:0] <= 6'b111111;
124             else if ( counter2 < 6'd9)
125                 leds[5:0] <= 6'b000000;
126             else if ( counter2 < 6'd10)
127                 leds[5:0] <= 6'b111111;
128             else begin
129                 leds[5:0] <= leds;
130                 i = i - 1;
131
132                 run=0;
133             end
134         end
135
136     end
```

```
137     "c": begin
138
139
140         if(!run && counter2 == 6'd0)
141             run=1;
142         if(run) begin
143             if ( counter2 < 6'd3)
144                 leds[5:0] <= 6'b000000;
145             else if ( counter2 < 6'd4)
146                 leds[5:0] <= 6'b111111;
147             else if ( counter2 < 6'd5)
148                 leds[5:0] <= 6'b000000;
149             else if ( counter2 < 6'd6)
150                 leds[5:0] <= 6'b111111;
151             else if ( counter2 < 6'd9)
152                 leds[5:0] <= 6'b000000;
153             else if ( counter2 < 6'd10)
154                 leds[5:0] <= 6'b111111;
155             else if ( counter2 < 6'd11)
156                 leds[5:0] <= 6'b000000;
157             else begin
158                 leds[5:0] <= leds;
159                 i = i - 1;
160
161                 run=0;
162             end
163
164
165
166
167     end
```

```
168     "d": begin
169         if(!run && counter2 == 6'd0)
170             run=1;
171         if(run) begin
172             if ( counter2 < 6'd3)
173                 leds[5:0] <= 6'b000000;
174             else if ( counter2 < 6'd4)
175                 leds[5:0] <= 6'b111111;
176             else if ( counter2 < 6'd5)
177                 leds[5:0] <= 6'b000000;
178             else if ( counter2 < 6'd6)
179                 leds[5:0] <= 6'b111111;
180             else if ( counter2 < 6'd7)
181                 leds[5:0] <= 6'b000000;
182             else if ( counter2 < 6'd8)
183                 leds[5:0] <= 6'b111111;
184
185             else begin
186                 leds[5:0] <= leds;
187                 i = i - 1;
188
189                 run=0;
190             end
191         end
192     end
193
194
195
196
197
```

```
198     "e": begin
199
200
201
202         if(!run && counter2 == 6'd0)
203             run=1;
204         if(run) begin
205             if ( counter2 < 6'd1)
206                 leds[5:0] <= 6'b000000;
207             else if ( counter2 < 6'd2)
208                 leds[5:0] <= 6'b111111;
209             else begin
210                 leds[5:0] <= leds;
211                 i = i - 1;
212
213                 run=0;
214             end
215
216
217
218
219     end
220
```

```
223      "s": begin
224          if(!run && counter2 == 6'd0)
225              run=1;
226          if(run) begin
227              if ( counter2 < 6'd1)
228                  leds[5:0] <= 6'b000000;
229              else if ( counter2 < 6'd3)
230                  leds[5:0] <= 6'b111111;
231              else if ( counter2 < 6'd4)
232                  leds[5:0] <= 6'b000000;
233              else if ( counter2 < 6'd6)
234                  leds[5:0] <= 6'b111111;
235              else if ( counter2 < 6'd7)
236                  leds[5:0] <= 6'b000000;
237              else if ( counter2 < 6'd9)
238                  leds[5:0] <= 6'b111111;
239          else begin
240              leds[5:0] <= leds;
241              i = i - 1;
242
243              run=0;
244          end
245      end
246
247
248
249      end
250
```

```
252      "l": begin
253          if(!run && counter2 == 6'd0)
254              run=1;
255          if(run) begin
256              if ( counter2 < 6'd1)
257                  leds[5:0] <= 6'b000000;
258              else if ( counter2 < 6'd2)
259                  leds[5:0] <= 6'b111111;
260              else if ( counter2 < 6'd5)
261                  leds[5:0] <= 6'b000000;
262              else if ( counter2 < 6'd6)
263                  leds[5:0] <= 6'b111111;
264              else if ( counter2 < 6'd7)
265                  leds[5:0] <= 6'b000000;
266              else if ( counter2 < 6'd9)
267                  leds[5:0] <= 6'b111111;
268              else if ( counter2 < 6'd10)
269                  leds[5:0] <= 6'b000000;
270              else if ( counter2 < 6'd11)
271                  leds[5:0] <= 6'b111111;
272          else begin
273              leds[5:0] <= leds;
274              i = i - 1;
275
276              run=0;
277          end
278
279
280
281
282      end
283
```

```
284      "m": begin
285
286
287
288
289          if(!run && counter2 == 6'd0)
290              run=1;
291          if(run) begin
292              if (counter2 < 6'd3)
293                  leds[5:0] <= 6'b000000;
294              else if ( counter2 < 6'd4)
295                  leds[5:0] <= 6'b111111;
296              else if ( counter2 < 6'd7)
297                  leds[5:0] <= 6'b000000;
298          else if ( counter2 < 6'd8)
299              leds[5:0] <= 6'b111111;
300
301
302
303
304
305
306          else begin
307              leds[5:0] <= leds;
308              i = i - 1;
309
310
311
312
313
314      endmodule
315
```

S

E

L

A

M

