

```
1 // 100 Days of RTL
2
3 // Abilash P
4
5 // Vending Machine
6
7 module vending_machine(clk, rst,
8                         select_snack_or_drink,
9                         select_snack,
10                        select_drink,
11                        money_inp,
12                        off_val,
13                        snack_or_drink_out,
14                        snack_out,
15                        drink_out,
16                        s_chips_avail,
17                        s_cookies_avail,
18                        s_chocolate_avail,
19                        s_icecream_avail,
20                        d_cocacola_avail,
21                        d_juice_avail,
22                        d_coffee_avail,
23                        d_7up_avail,
24                        avail_out,
25                        balance_out,
26                        dispense_out,
27                        stop_out,
```

```
28         over_out,  
29         refill_out);  
30  
31 parameter IDLE_STATE      = 4'b0000;  
32 parameter SELECT_1       = 4'b0001;  
33 parameter SELECT_2       = 4'b0010;  
34 parameter SELECT_3       = 4'b0011;  
35 parameter AVAIL_CHECK    = 4'b0100;  
36 parameter AMOUNT_COLLECT = 4'b0101;  
37 parameter BALANCE_STATE  = 4'b0111;  
38 parameter DISPENSE_STATE = 4'b1000;  
39 parameter STOP_STATE     = 4'b1001;  
40 parameter OVER_STATE     = 4'b1010;  
41 parameter REFILL_STATE   = 4'b1011;  
42  
43 parameter SNACK_CHIPS     = 10;  
44 parameter SNACK_COOKIES  = 15;  
45 parameter SNACK_CHOCOLATE= 20;  
46 parameter SNACK_ICECREAM = 40;  
47  
48 parameter DRINK_COACOLA  = 30;  
49 parameter DRINK_JUICE    = 25;  
50 parameter DRINK_COFFEE   = 15;  
51 parameter DRINK_7UP      = 25;  
52  
53 parameter SNACK_CHIPS_AVAIL    = 5;  
54 parameter SNACK_COOKIES_AVAIL = 3;  
55 parameter SNACK_CHOCOLATE_AVAIL= 5;  
56 parameter SNACK_ICECREAM_AVAIL= 3;  
57  
58 parameter DRINK_COACOLA_AVAIL = 3;  
59 parameter DRINK_JUICE_AVAIL   = 5;  
60 parameter DRINK_COFFEE_AVAIL  = 3;  
61 parameter DRINK_7UP_AVAIL     = 5;  
62
```

```

63 input clk, rst, select_snack_or_drink;
64 input [1:0]select_snack, select_drink;
65 input [5:0]money_inp;
66
67 output reg off_val, dispense_out;
68 output reg [1:0]snack_or_drink_out;
69 output reg avail_out, stop_out, over_out, refill_out;
70 output reg [1:0]snack_out, drink_out;
71 output reg [5:0]balance_out;
72
73 output reg [2:0]s_chips_avail, s_cookies_avail;
74 output reg [2:0]s_chocolate_avail;
75 output reg [2:0]s_icecream_avail;
76 output reg [2:0]d_cocacola_avail, d_juice_avail;
77 output reg [2:0]d_coffee_avail;
78 output reg [2:0]d_7up_avail;
79
80 reg [3:0]state, next_state;
81
82 always @(posedge clk)
83     begin
84         if(rst)
85             begin
86                 s_chips_avail    <=SNACK_CHIPS_AVAIL;
87                 s_cookies_avail  <=SNACK_COOKIES_AVAIL;
88                 s_chocolate_avail<=SNACK_CHOCOLATE_AVAIL;
89                 s_icecream_avail <=SNACK_ICECREAM_AVAIL;
90
91                 d_cocacola_avail <=DRINK_COCACOLA_AVAIL;
92                 d_juice_avail    <=DRINK_JUICE_AVAIL;
93                 d_coffee_avail   <=DRINK_COFFEE_AVAIL;
94                 d_7up_avail      <=DRINK_7UP_AVAIL;
95                 state            <=IDLE_STATE;
96             end
97

```

```

98     else
99         state<=next_state;
100     end
101
102 always @(state or select_snack_or_drink)
103     begin
104         case(state)
105             IDLE_STATE: begin
106                 off_val          <= 1;
107                 snack_or_drink_out <= 0;
108                 snack_out         <= 0;
109                 drink_out         <= 0;
110                 avail_out         <= 0;
111                 balance_out       <= 0;
112                 dispense_out      <= 0;
113                 stop_out          <= 0;
114                 over_out          <= 0;
115                 refill_out        <= 0;
116
117                 next_state<= SELECT_1;
118             end
119
120             SELECT_1: begin
121                 off_val          <= 0;
122                 snack_or_drink_out <= select_snack_or_drink;
123                 snack_out         <= 0;
124                 drink_out         <= 0;
125                 avail_out         <= 0;
126                 balance_out       <= 0;
127                 dispense_out      <= 0;
128                 stop_out          <= 0;
129                 over_out          <= 0;
130                 refill_out        <= 0;
131
132                 if(select_snack_or_drink==1)

```



```
133     next_state <= SELECT_2;    //snack state
134     else
135         next_state <= SELECT_3;    //drink state
136 end
137
138 SELECT_2: begin
139     off_val          <= 0;
140     snack_or_drink_out <= 0;
141     snack_out        <= select_snack;
142     drink_out         <= 0;
143     balance_out       <= 0;
144     avail_out         <= 0;
145     dispense_out      <= 0;
146     stop_out          <= 0;
147     over_out          <= 0;
148     refill_out        <= 0;
149
150     next_state<=AVAIL_CHECK;
151 end
152
153 SELECT_3: begin
154     off_val          <= 0;
155     snack_or_drink_out <= 0;
156     drink_out        <= select_drink;
157     snack_out        <= 0;
158     balance_out       <= 0;
159     avail_out         <= 0;
160     dispense_out      <= 0;
161     stop_out          <= 0;
162     over_out          <= 0;
163     refill_out        <= 0;
164
165     next_state<=AVAIL_CHECK;
166 end
167
```

```

168     AVAIL_CHECK: begin
169         off_val      <= 0;
170         snack_or_drink_out <= 0;
171         snack_out     <= 0;
172         drink_out     <= 0;
173         balance_out   <= 0;
174         dispense_out  <= 0;
175         stop_out      <= 0;
176         over_out      <= 0;
177         refill_out    <= 0;
178
179         if(s_chips_avail == 0 || s_cookies_avail == 0 || s_chocolate_avail ==
0 || s_icecream_avail == 0 || d_cocacola_avail == 0 || d_juice_avail == 0 ||
d_coffee_avail == 0 || d_7up_avail == 0)
180             begin
181                 avail_out <= 0;
182                 next_state <= OVER_STATE;
183             end
184
185         else
186             begin
187                 avail_out <= 1;
188                 if(select_snack_or_drink == 1)
189                     begin
190                         case(select_snack)
191                             0: s_chips_avail <= s_chips_avail - 1;
192                             1: s_cookies_avail <= s_cookies_avail - 1;
193                             2: s_chocolate_avail <= s_chocolate_avail - 1;
194                             3: s_icecream_avail <= s_icecream_avail - 1;
195                         endcase
196                     end
197                 else
198                     begin
199                         case(select_drink)
200                             0: d_cocacola_avail <= d_cocacola_avail - 1;

```

```
201         1: d_juice_avail <= d_juice_avail - 1;
202         2: d_coffee_avail <= d_coffee_avail - 1;
203         3: d_7up_avail <= d_7up_avail - 1;
204     endcase
205 end
206     next_state <= AMOUNT_COLLECT;
207 end
208 end
209
210 AMOUNT_COLLECT: begin
211     off_val <= 0;
212     snack_or_drink_out <= 0;
213     snack_out <= 0;
214     drink_out <= 0;
215     avail_out <= 0;
216     balance_out <= 0;
217     dispense_out <= 0;
218     stop_out <= 0;
219     over_out <= 0;
220     refill_out <= 0;
221
222     if(select_snack_or_drink==1) //1 for snack
223     begin
224         if(select_snack==0)
225         begin
226             if(money_inp==SNACK_CHIPS)
227                 next_state<=DISPENSE_STATE;
228             else if(money_inp>SNACK_CHIPS)
229                 next_state<=BALANCE_STATE;
230             else
231                 next_state<=STOP_STATE;
232         end
233
234         else if(select_snack==1)
235         begin
```

```
236         if(money_inp==SNACK_COOKIES)
237             next_state<=DISPENSE_STATE;
238         else if(money_inp>SNACK_COOKIES)
239             next_state<=BALANCE_STATE;
240         else
241             next_state<=STOP_STATE;
242     end
243
244     else if(select_snack==2)
245     begin
246         if(money_inp==SNACK_CHOCOLATE)
247             next_state<=DISPENSE_STATE;
248         else if(money_inp>SNACK_CHOCOLATE)
249             next_state<=BALANCE_STATE;
250         else
251             next_state<=STOP_STATE;
252     end
253
254     else
255     begin
256         if(money_inp==SNACK_ICECREAM)
257             next_state<=DISPENSE_STATE;
258         else if(money_inp>SNACK_ICECREAM)
259             next_state<=BALANCE_STATE;
260         else
261             next_state<=STOP_STATE;
262     end
263 end
264
265 else
266 begin
267     if(select_drink==0)
268     begin
269         if(money_inp==DRINK_COACOLA)
270             next_state<=DISPENSE_STATE;
```



```
271         else if (money_inp > DRINK_COACOLA)
272             next_state <= BALANCE_STATE;
273         else
274             next_state <= STOP_STATE;
275     end
276
277     else if (select_drink == 1)
278     begin
279         if (money_inp == DRINK_JUICE)
280             next_state <= DISPENSE_STATE;
281         else if (money_inp > DRINK_JUICE)
282             next_state <= BALANCE_STATE;
283         else
284             next_state <= STOP_STATE;
285     end
286
287     else if (select_drink == 2)
288     begin
289         if (money_inp == DRINK_COFFEE)
290             next_state <= DISPENSE_STATE;
291         else if (money_inp > DRINK_COFFEE)
292             next_state <= BALANCE_STATE;
293         else
294             next_state <= STOP_STATE;
295     end
296
297     else
298     begin
299         if (money_inp == DRINK_7UP)
300             next_state <= DISPENSE_STATE;
301         else if (money_inp > DRINK_7UP)
302             next_state <= BALANCE_STATE;
303         else
304             next_state <= STOP_STATE;
305     end
```

```
306     end
307 end
308
309 BALANCE_STATE: begin
310     off_val      <= 0;
311     snack_or_drink_out <= 0;
312     snack_out     <= 0;
313     drink_out     <= 0;
314     avail_out     <= 0;
315     dispense_out  <= 0;
316     stop_out      <= 0;
317     over_out      <= 0;
318     refill_out    <= 0;
319
320     if(select_snack_or_drink==1) //1 for snack
321     begin
322         if(select_snack==0)
323             balance_out<=money_inp-SNACK_CHIPS;
324         else if(select_snack==1)
325             balance_out<=money_inp-SNACK_COOKIES;
326         else if(select_snack==2)
327             balance_out<=money_inp-SNACK_CHOCOLATE;
328         else
329             balance_out<=money_inp-SNACK_ICECREAM;
330     end
331
332     else
333     begin
334         if(select_drink==0)
335             balance_out<=money_inp-DRINK_COCACOLA;
336         else if(select_drink==1)
337             balance_out<=money_inp-DRINK_JUICE;
338         else if(select_drink==2)
339             balance_out<=money_inp-DRINK_COFFEE;
340         else
```

```
341     balance_out<=money_inp-DRINK_7UP;
342     end
343     next_state<=DISPENSE_STATE;
344 end
345
346 DISPENSE_STATE: begin
347     off_val          <= 0;
348     snack_or_drink_out <= 0;
349     snack_out        <= 0;
350     drink_out        <= 0;
351     avail_out        <= 0;
352     balance_out      <= 0;
353     dispense_out     <= 1;
354     stop_out         <= 0;
355     over_out         <= 0;
356     refill_out       <= 0;
357
358     next_state<=SELECT_1;
359 end
360
361 STOP_STATE: begin
362     off_val          <= 0;
363     snack_or_drink_out <= 0;
364     snack_out        <= 0;
365     drink_out        <= 0;
366     avail_out        <= 0;
367     balance_out      <= 0;
368     dispense_out     <= 0;
369     stop_out         <= 1;
370     over_out         <= 0;
371     refill_out       <= 0;
372
373     if(select_snack_or_drink == 1)
374     begin
375         case(select_snack)
```

```

376         0: s_chips_avail <= s_chips_avail + 1;
377         1: s_cookies_avail <= s_cookies_avail + 1;
378         2: s_chocolate_avail <= s_chocolate_avail + 1;
379         3: s_icecream_avail <= s_icecream_avail + 1;
380     endcase
381 end
382
383 else
384     begin
385         case(select_drink)
386             0: d_cocacola_avail <= d_cocacola_avail + 1;
387             1: d_juice_avail <= d_juice_avail + 1;
388             2: d_coffee_avail <= d_coffee_avail + 1;
389             3: d_7up_avail <= d_7up_avail + 1;
390         endcase
391     end
392     next_state<=AVAIL_CHECK;
393 end
394
395 OVER_STATE: begin
396     off_val <= 0;
397     snack_or_drink_out <= 0;
398     snack_out <= 0;
399     drink_out <= 0;
400     avail_out <= 0;
401     balance_out <= 0;
402     dispense_out <= 0;
403     stop_out <= 0;
404     over_out <= 1;
405     refill_out <= 0;
406
407     next_state<=REFILL_STATE;
408 end
409
410 REFILL_STATE: begin

```



```
411 off_val      <= 0;
412 snack_or_drink_out <= 0;
413 snack_out      <= 0;
414 drink_out      <= 0;
415 avail_out      <= 0;
416 balance_out    <= 0;
417 dispense_out   <= 0;
418 stop_out       <= 0;
419 over_out       <= 0;
420 refill_out     <= 1;
421
422 if(select_snack_or_drink==1) //1 for snack
423     begin
424         if(select_snack==0)
425             begin
426                 avail_out <= 1;
427                 s_chips_avail <= s_chips_avail+1;
428             end
429
430         else if(select_snack==1)
431             begin
432                 avail_out <= 1;
433                 s_cookies_avail <= s_cookies_avail+1;
434             end
435
436         else if(select_snack==2)
437             begin
438                 avail_out <= 1;
439                 s_chocolate_avail <= s_chocolate_avail+1;
440             end
441
442         else
443             begin
444                 avail_out <= 1;
445                 s_icecream_avail <= s_icecream_avail+1;
```

```
446         end
447     end
448
449     else
450         begin
451             if(select_drink==0)
452                 begin
453                     avail_out  <= 1;
454                     d_cocacola_avail <= d_cocacola_avail+1;
455                 end
456
457             else if(select_drink==1)
458                 begin
459                     avail_out  <= 1;
460                     d_juice_avail <= d_juice_avail+1;
461                 end
462
463             else if(select_drink==2)
464                 begin
465                     avail_out  <= 1;
466                     d_coffee_avail <= d_coffee_avail+1;
467                 end
468
469             else
470                 begin
471                     avail_out  <= 1;
472                     d_7up_avail <= d_7up_avail+1;
473                 end
474             end
475             next_state<=SELECT_1;
476         end
477     endcase
478 end
479 endmodule
```

Get Signals   Radix ▾   🔍 🔍   100%   ⏮ ⏭   ⚡ 0ns   ⬆ ⬇ ✖

