You are to construct a digital circuit that plays a game of roulette, allows betting and keeps track of total earnings. The roulette wheel has 8 slots, labeled 1 . . . 8. The player can play one of the numbers straight or play even or odd. The player starts with $10. The layout of the machine is shown in Figure 1.

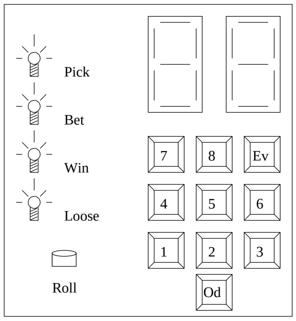


Figure 1: The layout of the roulette playing machine. The two 7 segment displays at the top are used for a variety of purposes.

The sequence of events is as follows:

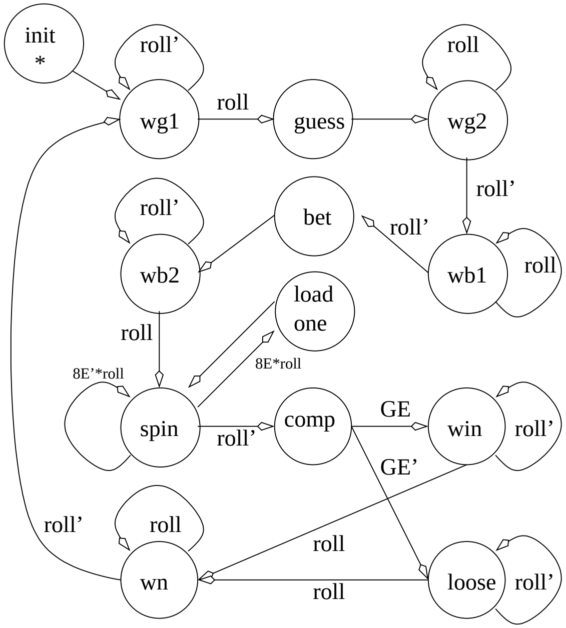
1. The circuit lights up the PICK LED. The player enters their guess; a number between 1-8, even or odd. While holding down their guess they press the roll button.
2. The circuit displays the picked number in the left most 7 segment display. The circuit lights up the BET LED. The player enters a one digit bet between 1 to 8. While holding down their bet they press the roll button.
3. The circuit displays the bet on the rightmost 7-segment display. The player pushes and holds down the roll button. The circuit increments a mod 8 counter while the roll button is depressed. It would be nice to display the current count value on right 7-seven segment display. Since the clock cycle is on the order of milliseconds, then the user would not be able to anticipate the roll.
4. The player releases the roll button. The final roll is displayed on the rightmost 7-segment display. The circuit stops incrementing the counter and checks to see if the final value matches the players guess. If the match is correct then light the WIN LED and increment the players earnings. If the match is incorrect then light the LOOSE LED and decrement the players earnings.
5. The play hits the roll button to clear the roll information from the 7-segment displays.
6. The circuit displays the players earnings on the 7-segment display.
7. When the user pushes the roll button then goto step 1.

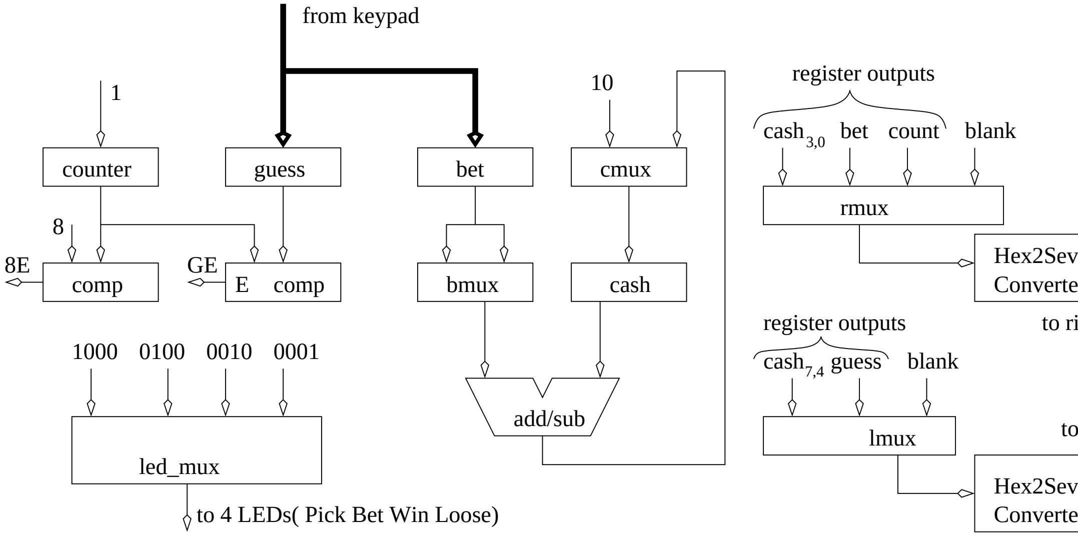
Set reasonable bounds on the maximum winnings. Values may be displayed in hexadecimal (you may assume that you have a hex to 7-segment display converter at your disposal).

Algorthm

```
cash = 10;
LeftSeven = blank;
RightSeven = cash;
while(cash ¿ 0)
LEDarray = 1 0 0 0; // Pick Bet Win Loose
while(ROLL == 0);
guess = Datain;
while(ROLL == 1);

LEDarray = 0 1 0 0; // Pick Bet Win Loose
LeftSeven = guess;
RightSeven = blank;
while(ROLL == 0);
bet = Datain;
while(ROLL == 1);

LEDarray = 0 0 0 0; // Pick Bet Win Loose
RightSeven = bet;
while(ROLL == 0);
while(ROLL == 1) count = count + 1;

RightSeven = count;
if (count == guess)
cash = cash + (bet ¡¡ 2);
LEDarray = 0 0 1 0; // Pick Bet Win Loose
 else
cash = cash - bet;
LEDarray = 0 0 0 1; // Pick Bet Win Loose

while(ROLL == 0);
while(ROLL == 1);
RightSeven = cash;


LEDarray = 0 0 0 1; // The user is out of money
while(1); // Halt the machine
```

Datapath and Control





Control Word

State	counter	guess	bet	cmux	bmux	cash	add/sub	ledmux	rmux	lmux
	00 hold	0 hold	0 hold	0 $10	0 bet	0 hold	0 add	00 (pick)	00 cash	00 cash
	01 load	0 load	1 load	1 add/sub	1 bet¡¡2	1 load	1 sub	01 (bet)	01 bet	01 guess
	10 up							10 (win)	10 count	10 blank
								11 (loose)	11 blank	
init										
wg1										
guess										
wg2										
wb1										
bet										
wb2										
spin										
load 1										
comp										
win										
loose										
wn										