HW4 Answer Keys/Tips

Q1

One logic expression has two product terms and others have three product terms.

Here is the output of the program on another string. Test your program with more random bit strings and decide yourself if the state machine works correctly. See the explanations below the sample output.

```
python q1.py 110101
state b | ns z v
  0   1 | 1  0 1
  1   1 | 0  1 3
  0   0 | 0  1 6
  0   1 | 1  0 13
  1   0 | 2  0 26
  2   1 | 2  0 53
  2   1 | 2  0 107
```

The state machine detects if a binary number of an arbitrary length is divisible by 3. The bits in the number are fed into the machine from left to right, i.e., from the most significant bit (the left-most bit) to the least significant bit, one bit per clock cycle. `z` is 1 if and only if the bits the state machine has seen (including the current input b) are a multiple of 3. So z should be 1 if and only if v is divisible by 3.

Although the state table is given in this question, it is not hard to construct one from scratch. We only need to keep track of the remainder (from dividing the number by 3) in the state machine, and update the state with the new remainder. When dividing a number by 3, we get 3 possible remainders: 0, 1, or 2. S0 is for remainder 0, S1 for 1, and S2 for 2. The new remainder/state is (state *2 + b) mod 3. S3 is treated the same as S0. No state transits into S3.

Although it is not required, you can generate the trace file (a VCD file) with `--trace` option on the command line and observe the waveforms of signals.

Q2

The figure in 2.a should be helpful for you to answer other questions.

The answer to 2.c is 87.0 MHz.

The answer to 2.d is 666.7 MHz. Study the figure in 2.a and think about how you can reduce the cycle time/increase the clock rate.

Q3

The product is 0b0111001011.

For two's complement numbers, find the product in decimal, convert it to a 2's complement number (i.e., find its representation in 2's complement number), and compare the lower halves.

Q4

Check a) the code can print correct strings, and b) registers are saved/restored properly in the function.

Compare the algorithm we (humans) convert a decimal number to a binary number.

Q5

a. 2.6
b. 3.3
c. 1.18
d. 2.25
e. 1.22

Q6

a. 1.25
b. 1.44
c. 2.012
d. 1.1044