ICS

SHEET 10

Arnav Singh

#	Machine Code	Assembly code	Description
0	001 1 0001	Load #1	Load value 1 in accumalator
1	010 0 1111	STORE 15	Store accumaltor val in 15th cell
2	001 1 0000	LOAD #0	Load value 0 in acc.
3	101 1 0100	Equal #4	Skip the next intsruction if the value of accumator equals 4.
4	110 1 0110	JUMP #6	Jump to instruction 6.
5	111 1 0000	Halt	STOP EXECUTION
6	001 0 0011	LOAD 3	Load the value of cell 3 into the accumalator
7	100 1 0001	SUB #1	Subtract value one from accum.
8	010 0 0011	STORE 3	Store value of accumalter into memory cell 3
9	001 0 1111	LOAD 15	load the value of memory location 15 into accumalater
10	011 0 1111	ADD 15	Add the value of memory location 15 into accumalater
11	010 0 1111	STORE 15	Store value of accumalator into memory location 15
12	110 1 0010	JUMP #2	JUMP to instruction 2
13	000 0 0000	—	no instuction
14	000 0 0000	—	—
15	000 0 000	—	—

b) The programm start with loading the value 1 into ~~accumalator~~ memory cell 15. It then loads 0 into Acc. Then it checks if 0 = 4, which is false so it wont skip the next instruction, i.e ~~~~ it now JUMP's to instruction 6. Now it loads the value of memory cell 3 into accumalater which is 4, value one now gets subtracted from 4. So we have 3 now being stored in memory cell 3. 'Equal #4' becomes 'Equal #3'

Now LOAD 15 loads 1 into accumalator which we got from instruction #0 & #1.
Add 15 adds the value 1 to itself so it is doubled and 2 is now stored back into cell 15.
Now Jump #2 goes back to instuction 2 and loads 2 into accumalator.
The same process is repeated but now we have Equal #3 so after every iteration (1 cycle) Equal #4
it goes down to Equal #0
Eventually Load 0 = Equal #0 so it stops by this time we now had numbers 2, 4, 8, 16 in memory location 15.
Therefore 16 is the result left in cell 15.

Equal #4 → Equal #3 → Equal #2 ... Equal #0
LOAD 0 = Equal #0
↳ HALT

2, 4, 8, 16

c) Mathematical expression :

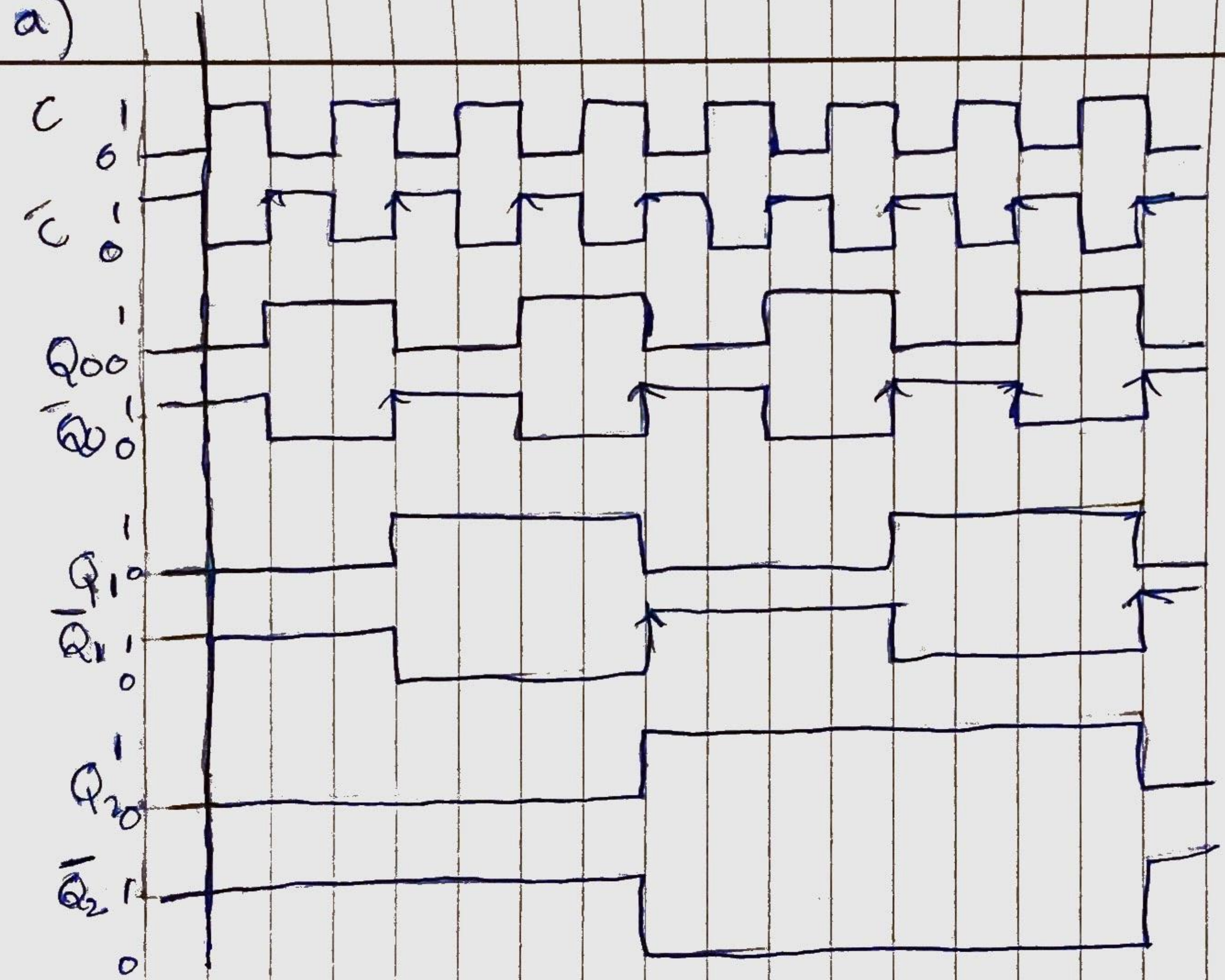
$$= \boxed{2n, \text{ wher } n = 1, 2, 4, 8}$$

or

$$\boxed{= 2^n \quad \text{where } n = 1, 2, 3, 4}$$

10.2

a)



b) No, you can't have arbitrarily long number of D Flip flops because there will be a time delay each time we add flip flops. Here we assumed no time delays but in real life time delays can have significant impact.