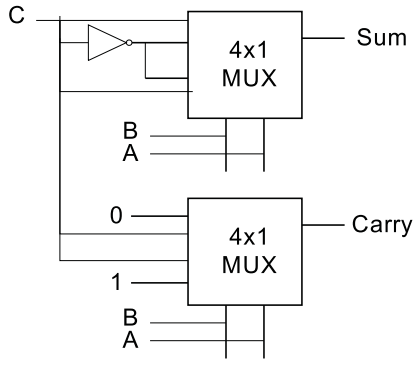
Esercizio 2. *(Punti 3)*
Il tempo T necessario a traferire un settore di $512 = 2^9$ byte is calcola sommando i tempi di seek T_s , di rotazione T_r e di trasferimento T_t:

$$\begin{aligned} T &= T_s + T_r + T_t = 10 \cdot 10^{-3}s + \frac{1}{2} \cdot \frac{60}{5000}s + \frac{2^9}{2^6} \cdot 10^{-6}s \\ &= T_s + T_r + T_t = 10 \cdot 10^{-3}s + 6 \cdot 10^{-3} + 8 \cdot 10^{-6}s \\ &\approx T_s + T_r + T_t = 10 \cdot 10^{-3}s + 6 \cdot 10^{-3} + 8 \cdot 10^{-6}s \end{aligned}$$

Esercizio 3. *(Punti 4)*

A	*B*	*C*	*Sum*	*Carry*
0	0	0	0	0
0	0	1	1	0
0	1	0	1	0
0	1	1	0	1
1	0	0	1	0
1	0	1	0	1
1	1	0	0	1
1	1	1	1	1

A	*B*	*Sum*	*Carry*
0	0	C	0
0	1	$\overline{C}$	C
1	0	$\overline{C}$	C
1	1	C	1



Esercizio 4. *(Punti 4)*

$$\begin{aligned} Q(t+1) = J\overline{Q}(t) + \overline{K}Q(t) \quad &\Rightarrow \quad \overline{Q}(t+1) = \overline{J\overline{Q}(t) + \overline{K}Q(t)} \\ &\Rightarrow \quad \overline{Q}(t+1) = \overline{J\overline{Q}(t)} \cdot \overline{\overline{K}Q(t)} \\ &\Rightarrow \quad \overline{Q}(t+1) = (\overline{J} + Q(t)) \cdot (K + \overline{Q}(t)) \\ &\Rightarrow \quad \overline{Q}(t+1) = \overline{J}K + \overline{JQ}(t) + KQ(t) + Q(t)\overline{Q}(t) \\ &\Rightarrow \quad \overline{Q}(t+1) = \overline{J}K + \overline{JQ}(t) + KQ(t) \\ &\Rightarrow \quad \overline{Q}(t+1) = \overline{J}K(Q(t) + \overline{Q}(t)) + \overline{JQ}(t) + KQ(t) \\ &\Rightarrow \quad \overline{Q}(t+1) = \overline{J}KQ(t) + \overline{J}K\overline{Q}(t) + \overline{JQ}(t) + KQ(t) \\ &\Rightarrow \quad \overline{Q}(t+1) = \overline{J}KQ(t) + KQ(t) + \overline{J}K\overline{Q}(t) + \overline{JQ}(t) \\ &\Rightarrow \quad \overline{Q}(t+1) = KQ(t) + \overline{JQ}(t) \end{aligned}$$

Esercizio 6. *(Punti 6)*

abcp1	`PC=PC+1`
abcp2	`OPC=MBR;fetch`
abcp3	`H=CPP`
abcp4	`MAR=H+MBRU;rd`
abcp5	`MAR=SP=SP+1`
abcp6	`PC=PC+1;fetch`
abcp7	`H=OPC`
abcp8	`TOS=MDR=H+MDR; wr; goto Main1`

Esercizio 8. *(Punti 5)*

```
subcompito:
        PUSH BP
        MOV BP,SP
        MOV AX,4(BP)
        MOV BX,6(BP)
        CMP AX,BX
        JLE minore
        SUB AX,BX
        JMP fine
minore:
        SUB BX,AX
        MOV AX,BX
fine:
        POP BP
        RET
```