

Prova Finale - Reti Logiche (prof. William Fornaciari)

Corigliano Emilio 10627041 - 907936

A.A. 2020/21

1 Considerazioni e scelte implementative

- Non è necessario distinguere le varie righe, si può trattare la memoria semplicemente come un vettore di dimensione $N = R \cdot C$ da analizzare sequenzialmente
- Ogni byte in input all'indirizzo $mem[2+i]$ dovrà essere processato e l'output dovrà essere scritto in memoria all'indirizzo $mem[2+N+i]$, con $i \in [0, N)$

1.1 Fasi

- una prima fase nella quale si calcola la dimensione dell'immagine N da equalizzare. Questo valore è ottenuto moltiplicando $mem[0]$ e $mem[1]$.
- una seconda fase nella quale si cercano i MAX_PV e MIN_PV : si scorre una volta tutta l'immagine e si cercano contemporaneamente il valore massimo e minimo dei pixel presenti nell'immagine
- una terza fase nella quale si calcola lo $SHIFT_LEVEL$, usando un metodo a soglie

Listing 1: Generazione di soglie

```
for i in range(256):  
    print(str(i) + " :  $\lfloor$ " + str(8 - math.floor(math.log2(i+1))))
```

- una quarta fase che è la vera e propria elaborazione dell'immagine, nella quale si scorre pixel per pixel, elaborandolo e scrivendolo nella regione di output della memoria (input: $mem[2+i]$; output: $mem[2+N+i]$)

1.2 Registri

- *RegN*: memorizza il numero totale di byte che compongono l'immagine
- *RegMaxPL*: memorizza il livello massimo dei pixel presente nell'immagine
- *RegMinPL*: memorizza il livello minimo dei pixel presente nell'immagine

- *RegShiftLevel*: memorizza lo *SHIFT_LEVEL*, può essere computato a soglie dato che il delta value è l'unica variabile.
- un contatore che permetta di scorrere per tutto il vettore l'immagine