

1) Per i mondiali di calcio 2018, una compagnia di scommesse considera Brasile (B), Spagna (S), Francia (F) e Germania (G) le quattro maggiori candidate alla vittoria finale. Se una o più di queste squadre non raggiungerà le semifinali del torneo, la compagnia di scommesse potrà già essere certa dei seguenti guadagni: 900K€ (B), 800K€ (S), 600K€ (F) e 500K€ (G). Si identifichino tramite tabella di verità le possibili combinazioni di accesso alle semifinali delle quattro squadre (0=assente, 1=presente) che permetterebbero alla compagnia di scommesse di ottenere un guadagno certo totale di almeno 1500K€. Si determini e si semplifichi la corrispondente espressione booleana e si disegni il relativo circuito digitale.

Procedimento

Espressione booleana semplificata

Circuito digitale semplificato

2) Siano y e x le ultime due cifre della propria matricola in base 10 (esempio: matricola 3465 $\rightarrow x=5, y=6$). Si consideri una CPU dotata di una cache di 1° livello con tempo di accesso di $2+y$ nanosecondi, una di 2° livello con tempo di accesso di $4+y$ nanosecondi e una di 3° livello con tempo di accesso di $6+x+y$ nanosecondi. Durante l'esecuzione di un programma, la CPU ha impiegato 6,5 microsecondi per leggere dalla memoria $(35 + x + 2y)$ parole. Sapendo che, all'inizio dell'esecuzione del programma, $y+2$ di tali parole erano già contenute nella cache di 1° livello, $y+4$ in quella di 2° livello e $x+10$ in quella di 3° livello e che durante l'esecuzione non è mai stato necessario rimpiazzare parole nelle cache, si calcoli il tempo di accesso a una parola della RAM, esprimendo il risultato in nanosecondi.

Risultato (in ns)

Procedimento

3) Si trasformi in formato IEEE 754 singola precisione il numero ottenuto ponendo nelle caselle libere le due cifre meno significative della propria matricola e si fornisca il risultato in esadecimale. Il numero da convertire è negativo se la matricola è pari, positivo altrimenti.

Numero formato IEEE (in esadecimale)

±?	8			1	7	●	1	8	7	5	=						
----	---	--	--	---	---	---	---	---	---	---	---	--	--	--	--	--	--

4) Dato il seguente programma in linguaggio assembly, indicare (in esadecimale) il valore delle variabili Ris1, Ris2, Ris3, Ris4 e Ris5 al termine dell'esecuzione. Il programma è suddiviso in 5 frammenti tra loro indipendenti.

```
unsigned short int Mat=....;    // INSERIRE QUI LE 4 CIFRE MENO SIGNIFICATIVE DELLA  
                                PROPRIA MATRICOLA (trattandole come numero decimale)
```

```
unsigned char Vet [100];        // Vettore di 100 BYTE  
unsigned short Ris1,Ris2,Ris3,Ris4,Ris5;    // WORD
```

```
MOV AX,Mat  
MOV BX,AX  
AND BX,5AFFh  
INC BX  
SUB AX,BX  
MOV Ris1,AX
```

Ris1=

// -----

```
MOV DX,Mat  
XOR DX,0F0Fh  
MOV WORD PTR Vet[63],DX  
INC DX  
MOV WORD PTR Vet[31],DX  
MOV BX,Mat  
AND EBX,2h  
OR EBX,2h  
SHL EBX,5  
DEC EBX  
MOV AX,WORD PTR Vet[EBX]  
MOV Ris2,AX
```

Ris2=

// -----

```
MOV AX,Mat  
MOV BYTE PTR Vet[10],AL  
MOV BYTE PTR Vet[13],99  
MOV BYTE PTR Vet[16],AH  
LEA ESI,Vet  
ADD ESI,9  
XOR EBX,EBX  
MOV ECX,3  
11: XOR BL,[ESI+1]  
ADD ESI,3  
LOOP 11  
MOV Ris3,BX
```

Ris3=

// -----

```
MOV AX,Mat  
XOR AH,AH  
XOR BL,BL  
ADD BL,8  
DIV BL ; Divis. senza segno di AX per r/m8: risultato in AL, resto in AH  
MOV Ris4,AX
```

Ris4=

// -----

```
MOV AX,Mat  
AND AL,7Fh  
XOR BL,BL  
MOV BL,-4  
IMUL BL ; Moltiplic. con segno di AL per r/m8: ris. in AX  
ADD AX,0Fh  
MOV Ris5,AX
```

Ris5=

1) In una moderna CPU multi-core come sono generalmente organizzati i tre livelli di cache?

2) Descrivere il tipo di interfacciamento/BUS che utilizzano le schede grafiche odierne evidenziando come sia possibile raggiungere elevate ampiezze di banda.

3) Che cos'è la codifica dei caratteri UNICODE e per quale motivo è stata introdotta?

4) Quali sono le principali differenze fra un compilatore e un interprete? Perché un programma compilato è in genere più veloce di uno interpretato?

5) Che cos'è il processo di fotolitografia utilizzato nella fabbricazione dei circuiti integrati?

6) Definire le modalità “Big Endian” e “Little Endian” e mostrare come la DWORD (A00BB75Fh) viene memorizzata, in ciascuna delle due modalità, a partire dall'indirizzo di memoria 1024.