ESAME CALCOLATORI

1. latenza tipo R

Mem-I - Mem-D	250
Register File	150
Mux	25
ALU	200
Addizionatore	150
Porta AND	5
lettura registro (solo PC)	30
impostazione registro	20
estensione segno	50
controllo	50

tipo R = 30+250+150+25+200+25+20 = 700

2. una funzione foglia (funzione che non invoca altre funzioni)
   - non ha mai necessità di salvare sullo stack il contenuto di alcun registro
   - può essere sempre ottimizzata eliminando il prologo e l'epilogo standard
   - dipendentemente da architettura e ABI, deve necessariamente salvare sullo stack il contenuto di almeno un registro
   - deve necessariamente salvare sullo stack il contenuto di alcuni registri
   - nessuna delle altre risposte

3. $1000\ 0000\ 0000_2 + 19_{16}$

   $= 1000\ 0001\ 1001_2$

4. $x8 = 1101\ 0000\ 0000\ 0000\ 0000\ 1100\ 0000\ 0000$

   $slli\ x8, x8, 6 \rightarrow x8 = 0000\ 0000\ 0000\ 0011\ 0000\ 0000\ 0000\ 0000$
   $srli\ x5, x8, 24 \rightarrow x5 = 0000\ 0000\ 0000\ 0000\ 0000\ 0000\ 0000\ 0000$
   $or\ x8, x8, x5 \rightarrow x8 = 0000\ 0000\ 0000\ 0011\ 0000\ 0000\ 0000\ 0000$

   $x8 = 0000\ 0000\ 0000\ 0011\ 0000\ 0000\ 0000\ 0000$

5. cache associativa a 2 vie, 16 KB, 16 byte per blocco, numero blocco parola ad inidirizzo 0x100400

   num blocchi = 16 * 1024 / 16 = 1024 → set = log (1024 / vie) = log(512) = 9
   offset = 4

   indirizzo = 0001 0000 000 0 0100 0000 0000

   set → 2^6 → 64 → * num vie = blocco 128 o 129

6. in Intel: *addb* $1, *%rax*

   - è scorretta sintatticamente: “b” significa byte ma %rax ha ampiezza diversa
   - è scorretta sintatticamente: il prefisso corretto per il valore immediato deve essere #
   - è scorretta sintatticamente: l’argomento <dst> non può essere un valore immediato
   - somma 1 al contenuto di %rax
   - nessuna delle altre risposte

7. funzione (mi pare fosse puntatori u, v, z , for (i=0, i<size, i++) e z[i] = u[i]+v[i])

```
    testl %ecx, %ecx
    je .L1
    leal -1(%rcx), %eax
    leaq 8(, %rax, 8), %r8
    xorl %eax, %eax
.L3 movq (%rdi, %rax), %ecx
    addq (%rsi, %rax), %ecx
    movq %ecx, (%rdx, %rax)
    X1
    cmpq %rax, %r8
    X2
.L1 ret
```

   - X1 addl $4, %rax
     X2 je .L3
   - X1 addq $4, %rax
     X2 jne .L3
   - X1 addq $4, %rax
     X2 jle .L3
   - X1 movq $0, %rax
     X2 jle .L3
   - X1 addq (%rsi, %rax), %rax
     X2 movq (%rdi, %rax), %rax

8. qual è il formato corretto per i salti in RISC-V:

    - opcode - indirizzo
    - opcode - rs - indirizzo
    - opcode - rs- rt - indirizzo
    - opcode - rs - rt - rd -indirizzo
    - nessuna delle altre risposte

9. quale non è corretta in RISC-V:

    - $lb\ x5,\ 0(x10)$
    - $ld\ x5,\ (x10,\ x11)$
    - $ld\ x5,\ 8(x10)$
    - $lw\ x5,\ -12(x10)$
    - nessuna delle altre risposte

10. convertire il numero -38.03125 in binario con IEEE754

    = $1\ 10000100\ 00110000010000000000000$

11. calcolare -35_{10} in complemento a 2 su 8 bit

    $35\ =\ 0010\ 0011$

    CA1 $\rightarrow\ 1101\ 1100\ \rightarrow$ +1 $\rightarrow\ 1101\ 1101$

12. riguardo alla memory mapped I/O

    - il programmatore ha controllo diretto sulle locazioni di memoria
    - usa delle istruzioni LM speciali
    - può essere implementata usando qualsiasi locazione di memoria
    - è una modalità per impartire i comandi ai dispositivi
    - nessuna delle altre risposte

    risposta: è una delle due modalità per impartire comandi ai dispositivi fornendo sulle linee di bus alcune parole di controllo. Memory mapped I/O scrive/legge in particolari locazioni di memoria, l'altro modo è tramite alcune istruzioni

speciali dedicate all’I/O

es: scrivendo una particolare parola in una locazione di memoria associata al dispositivo → il sistema di memoria ignora la scrittura e il controllore di I/O intercetta l’indirizzo particolare e trasmette il dato sotto forma di comando

queste locazioni sono inaccessibili ai programmi utente, sono accessibili solo al SO

il dispositivo stesso può usare queste locazioni per trasmettere dati o pre-segnalare il suo stato