

Prima Prova Intermedia AESO (B)

2024/2025

Rispondere alle domande seguenti nel tempo limite di **30 minuti** dall'orario di pubblicazione. Sottomettere prima della deadline.

RACCOMANDAZIONE: si consiglia di evitare la sottomissione negli ultimissimi istanti in quanto la probabilità di qualche ritardo di rete o disconnessione per sovraccarico è molto alta; consegnare pertanto con qualche minuto di anticipo. Decidere di sottomettere negli ultimi secondi è una decisione di cui vi assumete la piena responsabilità: se non riuscite ad inviare il compito non potrà essere corretto e quindi non sarà valido.

1. Email *

2. Inserisci nel box sotto il tuo numero di matricola

3. In un registro a n bit con segnale di abilitazione indicare quale delle seguenti affermazioni è falsa

Contrassegna solo un ovale.

- L'uscita del registro è a n bit così come l'ingresso principale
- Il segnale di abilitazione è di 1 solo bit
- Il registro aggiorna la sua uscita con l'ingresso quando il clock è a 1 e il segnale di abilitazione è a 1
- Il segnale di abilitazione è posto in AND con il segnale di clock

4. In un Flip-Flip di tipo D indicare quale affermazione è vera tra le seguenti

Contrassegna solo un ovale.

- Il master è opaco durante la fase in cui il clock è a 1
- E' composto da un solo Latch D
- E' composto da due registri a 1 bit riceventi lo stesso clock
- E' composto da due Latch D riceventi lo stesso clock

5. Utilizzando la Mappa di Karnaugh seguente per minimizzare il più possibile la funzione Booleana, indicare quale affermazione è vera

	00	01	11	10
00	1	1	1	1
01	1	1	1	0
11	0	0	1	1
10	0	0	0	0

Contrassegna solo un ovale.

- La mappa non consente alcuna semplificazione
- Abbiamo cinque cerchi tutti di esattamente da due 1
- Abbiamo 5 implicanti di cui tre da quattro 1
- Abbiamo 4 implicanti di cui tre da quattro 1
- Abbiamo 4 implicanti di cui due da quattro 1

6. Dato un confrontatore con due ingressi a n bit realizzato secondo l'approccio "strutturale" con n confrontatori con due ingressi a 1 bit

Contrassegna solo un ovale.

- Il numero minimo di livelli di logica dell'implementazione "strutturale" è di 2
- Richiede una porta AND con n ingressi, quindi da realizzare con un numero di livelli di logica $\log_8(n)$ assumendo porte con al massimo 8 ingressi
- Il numero di livelli di logica cresce linearmente rispetto a n
- Il numero di livelli di logica è costante rispetto a n
- Richiede una porta AND con n ingressi, quindi da realizzare con un numero di livelli di logica $n/8$ assumendo porte con al massimo 8 ingressi

7. Quanti stati deve avere (almeno) un'automa di MOORE che riconosce una stringa di 3 'a' consecutive su un alfabeto $\{a,b,c\}$

Contrassegna solo un ovale.

- 3
- 4
- 5
- Nessuna delle precedenti

8. Due automi, uno di Moore e l'altro di Mealy, riconoscono la stessa stringa su un alfabeto $\{a,b\}$. Quale delle seguenti affermazioni è sicuramente vera?

Contrassegna solo un ovale.

- La rete che implementa l'automa di Moore ha un ciclo di clock più lungo di quella che implementa la rete di Mealy
- La rete che implementa l'automa di Moore ha un ciclo di clock minore o uguale di quella che implementa la rete di Mealy
- L'automa di Moore ha più stati interni del corrispondente automa di Mealy
- L'automa di Mealy ha più stati interni del corrispondente automa di Moore
- Nessuna delle precedenti affermazioni è sicuramente vera

9. Un multiplexer con 4 ingressi e un ingresso di controllo da 2 bit

Contrassegna solo un ovale.

- Può essere implementato con più multiplexer da 2 ingressi e un ingresso di controllo da 1 bit con lo stesso ritardo di stabilizzazione del multiplexer più piccolo
- Può essere implementato con più multiplexer da 2 ingressi e un ingresso di controllo da 1 bit con un ritardo di stabilizzazione pari a due volte quello del multiplexer più piccolo
- Non può essere implementato utilizzando multiplexer a due ingressi + ingresso di controllo da 1 bit
- Tutte le affermazioni precedenti sono false

10. Si considerino le espressioni Booleane in figura. Assumendo di avere a disposizione solo porte AND e OR da due ingressi, qual'è il numero minimo di livelli di logica necessario per calcolare una delle espressioni seguenti?

$$(\bar{a}b + a\bar{b})\bar{c}$$
$$\bar{a}\bar{b}c + a\bar{b}\bar{c}$$
$$\bar{a}b + \bar{b}c + a\bar{c}$$
$$(\bar{a}+\bar{b})\bar{c}$$

Contrassegna solo un ovale.

- 1
- 2
- 3
- 4
- 5

11. Per implementare una rete combinatoria che moltiplica due numeri interi senza segno da 2 bit per ottenere un risultato da 4 bit

Contrassegna solo un ovale.

- Dobbiamo prevedere un bit di overflow, da mettere a 1 quando il valore del prodotto non è rappresentabile con 4 bit
- Non serve il bit di overflow, visto che due valori rappresentati su 2 bit moltiplicati fra di loro non generano mai un risultato che non sia rappresentabile su 4 bit
- Non serve un bit per l'overflow, in quanto possiamo utilizzare a tale scopo il bit più significativo del risultato
- Serve il bit di overflow implementato tramite il carry

12. Una funzione pura che calcola un bit di risultato a partire da 8 bit di ingresso

Contrassegna solo un ovale.

- Può sempre essere calcolata utilizzando solo due livelli di logica, ovvero un unico livello di porte AND e un unico livello di porte OR (cioè un'unica porta OR)
- Anche utilizzando porte da 8 ingressi, non è detto che basti un solo livello di porte OR
- Anche utilizzando porte da 8 ingressi, non è detto che basti un solo livello di porte AND
- Il numero di livelli di logica è sicuramente $\log_2(8)$ ovvero 3

13. In un latch SR, qual'è la combinazione di valori in ingresso che non porta ad un risultato corretto delle uscite?

Contrassegna solo un ovale.

- SR = 00
- SR = 01
- SR = 10
- SR = 11
- Nessuna delle precedenti: qualunque sia il valore degli ingressi S ed R il circuito si stabilizza sempre in modo da fornire i valori in uscita Q e non(Q) corretti

14. L'implementazione di una funzione con stato, secondo il modello di Moore

Contrassegna solo un ovale.

- Richiede una rete combinatoria che calcola il prossimo stato interno e una rete combinatoria che calcola le uscite, entrambe diverse dalla rete che implementa la funzione identità
- Non richiede necessariamente un registro di stato interno
- Richiede necessariamente un registro di stato interno
- Richiede necessariamente sia le reti di cui alla prima risposta che il registro di stato

15. La tabella di verità di una funzione di 4 ingressi è rappresentata dalla mappa di Karnaugh in figura. Qual'è il numero minimo di implicant (termini AND) richiesti per la sua implementazione?

		ab			
		00	01	11	10
cd	00	1	0	0	1
	01	1	1	1	1
	11	0	1	1	0
	10	0	1	1	0

Contrassegna solo un ovale.

- 1
- 2
- 3
- 4
- 5
- 6
- 7
- 8

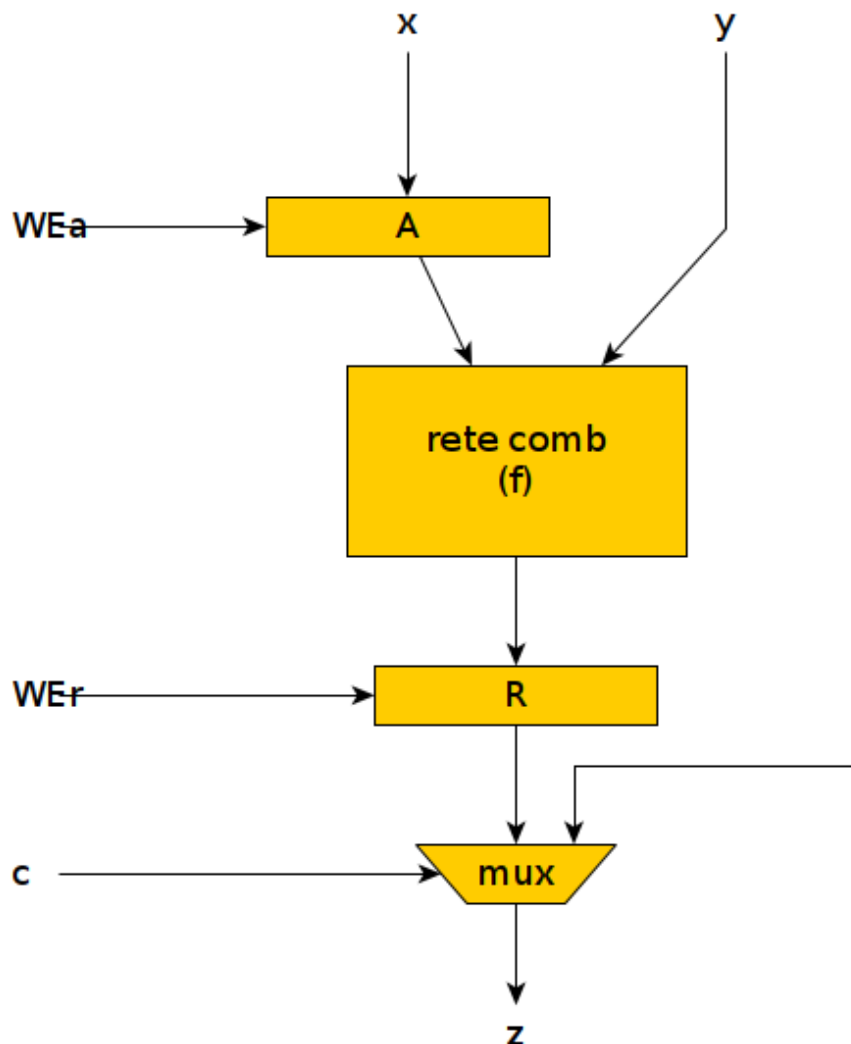
16. Come posso semplificare l'espressione in figura?

$$\bar{a}b + ab + a\bar{c}$$

Contrassegna solo un ovale.

- Posso solo raccogliere 'a' nei primi due termini AND e ridurre l'espressione ad "a OR (a AND (c negato))"
- Posso ridurre l'espressione ad 'a' (nessuna porta AND e nessuna porta OR richiesta)
- L'espressione è irriducibile
- Devo solo raccogliere 'a' negli ultimi due implicant

17. Quale delle seguenti affermazioni è vera?



Contrassegna solo un ovale.

- La rete in figura è una rete di Moore; lo stato interno è rappresentato dal registro R
- La rete in figura è una rete di Moore; lo stato interno è rappresentato dal registro R e dal registro A
- La rete in figura è una rete di Mealy; lo stato interno è rappresentato dal registro R
- La rete in figura è una rete di Mealy; lo stato interno è rappresentato dal registro R e dal registro A

18. Il modulo Verilog in figura

```
module m(output z, input x, input y, input z);  
  
    assign  
        z = (x && y) || (~x && z);  
  
endmodule
```

Contrassegna solo un ovale.

- Implementa la somma di due bit con un ingresso "riporto" da un altro modulo senza tenere conto dell'eventuale riporto generato
- Implementa un multiplexer da 2 ingressi (più ingresso di controllo)
- Implementa un demultiplexer da 2 uscite e un ingresso (più ingresso di controllo)
- Nessuna delle tre risposte è vera

19. Verilog utilizzato per il processo di "sintesi" di circuiti logici

Contrassegna solo un ovale.

- Permette di simulare il comportamento di una qualunque rete (combinatoria o sequenziale)
- Permette di simulare il comportamento di una qualunque rete sequenziale
- Permette di generare tutto quanto necessario ad implementare su silicio una qualunque rete (combinatoria o sequenziale)
- Permette di generare tutto quanto necessario ad implementare su silicio una qualunque rete sequenziale che implementi esclusivamente componenti di tipo processore

20. In un componente "primitive" di Verilog, la cui struttura è riportata in figura, le righe tra "table" e "endtable" non mostrate in figura (...)

```
primitive f(output z, input x, input y);  
  table  
    ...  
  endtable  
endprimitive
```

Contrassegna solo un ovale.

- Contengono uno o più statement "assign" che assegnano espressioni Booleane sulle variabili in ingresso alla variabile in uscita
- Contengono uno statement "assign" che assegna il risultato di una espressione Booleana sulle variabili in ingresso alla variabile in uscita
- Contengono valori binari delle variabili di input associati a valori binari della variabile di output
- Ogni riga è una configurazione di ingresso associata alla corrispondente configurazione di uno o più bit di uscita

21. Dati due moduli A e B in Verilog, che implementano componenti di logica combinatoria, posso creare un module Verilog che li usa entrambi per fornire un nuovo modulo che implementa logica combinatoria

Contrassegna solo un ovale.

- Utilizzando istanze dei moduli e dichiarazioni di tipo WIRE per creare i collegamenti fra i moduli componenti. Ogni wire connette un'uscita di uno dei moduli componenti ad un ingresso dell'altro modulo.
- Utilizzando istanze dei moduli e dichiarazioni di tipo WIRE per creare i collegamenti fra i moduli componenti. Ogni wire connette un'uscita di uno dei moduli componenti ad un ingresso dell'altro modulo oppure fra i valori in uscita dei moduli componenti e le variabili di uscita del nuovo modulo
- Utilizzando istanze dei moduli e dichiarazioni di tipo WIRE per creare i collegamenti fra i moduli componenti. Ogni wire connette un'uscita di uno dei moduli componenti ad un ingresso dell'altro modulo oppure fra i valori in ingresso dei moduli componenti e le variabili di ingresso del nuovo modulo.
- Nessuna delle due opzioni è una possibilità

22. Per il calcolo della lunghezza minima del ciclo di clock per far funzionare correttamente una rete sequenziale

Contrassegna solo un ovale.

- Se la rete è di Moore, occorre tenere in considerazione il ritardo di propagazione di Sigma soltanto, in quanto Omega dipende solo dallo stato interno
- Se di Moore, essendoci meno stati interni, si considera il ritardo di propagazione della sola Omega che richiede più tempo
- Si considera il massimo tra il ritardo di propagazione delle reti Sigma e Omega nel solo caso di rete di Mealy
- Si considera la somma dei ritardi di propagazione delle reti Sigma e Omega nel solo caso di Moore
- Nessuna delle precedenti

Questi contenuti non sono creati né avallati da Google.

Google Moduli

Terza Prova Intermedia AESO (B)

2024/2025

Rispondere alle domande seguenti nel tempo limite di **30 minuti** dall'orario di pubblicazione. Sottomettere prima della deadline.

RACCOMANDAZIONE: si consiglia di evitare la sottomissione negli ultimissimi istanti in quanto la probabilità di qualche ritardo di rete o disconnessione per sovraccarico è molto alta; consegnare pertanto con qualche minuto di anticipo. Decidere di sottomettere negli ultimi secondi è una decisione di cui vi assumete la piena responsabilità, se non riuscirete ad inviare il compito non potrà essere corretto e quindi non sarà valido.

* Indica una domanda obbligatoria

1. Email *

2. Numero di matricola *

3. Dire quali delle seguenti affermazioni è vera. In una memoria modulare interallacciata con 4 moduli ciascuno da 1K posizioni

Contrassegna solo un ovale.

- Gli indirizzi 0x0FE e 0x0FF sono indirizzi consecutivi dello stesso modulo da 1K
- Un indirizzo pari a un indirizzo dispari non sono mai indirizzi di posizioni nello stesso modulo da 1K
- Gli indirizzi 0x0F0 e 0x0F4 sono indirizzi di parole che si trovano su due moduli diversi
- Tutte le affermazioni precedenti sono false

4. Si consideri una memoria modulare sequenziale, strutturata con 4 moduli da 1K posizioni ciascuno, su cui viene effettuata una ricerca sequenziale di un valore in un array da 16 posizioni. Quale delle seguenti affermazioni è vera?

Contrassegna solo un ovale.

- può richiedere un accesso ad uno solo dei moduli, indipendentemente dall'indirizzo base del vettore, in caso di elemento trovato nelle prime 4 posizioni del vettore
- può richiedere un accesso ad uno solo dei moduli, in caso di elemento trovato nelle prime 4 posizioni del vettore, ma solo se l'indirizzo base del vettore è un multiplo di 4
- richiede comunque l'accesso a tutti e 4 i moduli di memoria, indipendentemente dall'indirizzo base del vettore e dalla posizione dell'elemento cercato nel vettore
- Nessuna delle precedenti è vera

5. In una memoria statica, organizzata come visto a lezione utilizzando un multiplexer per le letture e un demultiplexer per le scritture

Contrassegna solo un ovale.

- Il tempo di lettura è superiore o uguale al tempo di scrittura
- il tempo di scrittura è superiore o uguale al tempo di lettura
- il tempo di scrittura è uguale al tempo di lettura
- non possiamo dire nulla se non si conosce la dimensione della memoria (numero delle posizioni N)

6. Quale meccanismo si utilizza per ritardare l'applicazione dei segnali di controllo generati dalla parte controllo di un processore pipeline durante la fase di decodifica in modo che abbiano effetto al giusto ciclo di clock?

Contrassegna solo un ovale.

- reti combinatorie che introducono un ritardo di uno o più cicli di clock
- registri non architetturali che hanno come input i segnali di controllo dello stadio i e come uscita i segnali di controllo dello stadio $i+1$
- automi a stati finiti (modello di Moore) che ritardano le uscite di un ciclo di clock rispetto agli ingressi
- Nessuna delle precedenti risposte è vera

7. Quale parte dell'istruzione NON viene utilizzata per generare il segnale di controllo del multiplexer che sceglie l'ingresso del registro PC fra l'uscita di una ALU che calcola $+4$ e quella dell'ALU principale posta dopo il Register File nell'architettura pipeline?

Contrassegna solo un ovale.

- il campo FUNC
- il campo OP
- il campo SRC2
- il campo COND

8. Quale delle seguenti sequenze di istruzioni richiede un maggior numero di cicli di stallo in un processore pipeline?

Contrassegna solo un ovale.

- ADD R1, R2, R3 seguita da una SUB R5, R4, R1
- ADD R1, R2, R3 seguita da una SUB R5, R4, R4
- LDR R1, [R2], R3 seguita da una ADD R5, R3, R4
- LDR R1, [R2], R3 seguita da una ADD R5, R3, R1
- ADD R1, R2, R3 seguita da una STR R1, [R5], #4
- CMP R1, R2 seguita da una B loop
- SUBS R0, R0, R0 seguita da una BNE loop

9. Quale delle seguenti sequenze di istruzioni può essere eseguita nell'ordine di esecuzione inverso senza conseguenze sulla semantica?

Contrassegna solo un ovale.

- ADD R1, R2, R3 e SUB R4, R2, R3
- ADD R1, R2, R3 e SUB R1, R2, R3
- ADD R1, R2, R3 e SUB R3, R2, R1
- ADD R1, R2, R3 e SUB R2, R5, R6

10. Quale delle seguenti affermazioni è vera?

Contrassegna solo un ovale.

- Data una qualunque sequenza di istruzioni con una dipendenza logica nella forma LDR R1, [...] seguita da una ADD ..., ..., R1 può sempre essere riscritta spostando un'istruzione precedente alla LDR fra la LDR e la ADD
- Data una qualunque sequenza di istruzioni con una dipendenza logica nella forma LDR R1, [...] seguita da una ADD ..., ..., R1 può sempre essere riscritta spostando un'istruzione successiva alla ADD fra la LDR e la ADD
- Data una qualunque sequenza di istruzioni con una dipendenza logica nella forma LDR R1, [...] seguita da una ADD ..., ..., R1 può sempre essere riscritta cambiando la LDR con la coppia di istruzioni LDR R4, [...] e MOV R1, R4 e lasciando poi la ADD originale a seguire
- Nessuna delle affermazioni è vera

11. In un processore multi-cycle l'unità di controllo è una macchina a stati finiti

Contrassegna solo un ovale.

- Con un numero di stati pari al numero di istruzioni di tipo diverso interpretate dalla micro-architettura (operative, salti e istruzioni di memoria)
- Con un numero di stati pari al numero degli stadi del pipeline (fetch, decode, execute, data-memory, writeback)
- Con un numero di stati pari al numero del tipo di istruzioni per il numero di stadi del pipeline (3*5)
- Nessuna delle precedenti risposte è vera

12. In quale micro-architettura, fra quelle viste a lezione, la memoria istruzioni è distinta dalla memoria dati?

Contrassegna solo un ovale.

- Solo nella micro-architettura single-cycle
- Solo nella micro-architettura multi-cycle
- Nella micro-architettura single-cycle e in quella pipeline
- Nella micro-architettura single-cycle e in quella multi-cycle
- Nella micro-architettura multi-cycle e in quella pipeline

13. Il Register File nell'architettura single-cycle

Contrassegna solo un ovale.

- Ha due indirizzi di lettura e uno di scrittura, sufficienti a completare l'esecuzione di una qualunque istruzione Assembler in un ciclo di clock
- Ha due indirizzi di lettura e uno di scrittura, sufficienti a completare l'esecuzione di una qualunque istruzione Assembler in un ciclo di clock diversa da una istruzione STR Rx, [Ry, Rz]
- Ha due indirizzi di lettura e uno di scrittura, sufficienti a completare l'esecuzione di una qualunque istruzione Assembler in un ciclo di clock diversa da una istruzione STR Rx, [Ry, #4]
- Ha due indirizzi di lettura e uno di scrittura, sufficienti a completare l'esecuzione di una qualunque istruzione assembler in un ciclo di clock diversa da una istruzione STR Rx, [Ry], #4

14. Per implementare il forwarding che permette di eseguire due istruzioni operative consecutive con una dipendenza logica fra la prima e la seconda senza cicli di stallo

Contrassegna solo un ovale.

- Si introduce nel datapath un collegamento fra l'uscita del registro non-architetturale che segue la ALU e l'ingresso del registro non-architetturale che la precede, mediato da un multiplexer comandato dalla parte controllo
- Si introduce nel datapath un collegamento fra l'uscita del registro non-architetturale che segue la ALU e l'ingresso della ALU, mediato da un multiplexer comandato dalla parte controllo
- Si introduce nel datapath un collegamento fra l'uscita della ALU e l'ingresso della ALU, mediato da un multiplexer comandato dalla parte controllo
- Si introduce un collegamento fra l'uscita del registro non-architetturale che segue la ALU e quello che precede l'unità registri (fase di decode), mediato da un multiplexer comandato dalla parte controllo

15. Supponendo di avere un ciclo di clock pari a 5 nsec per la micro-architettura single-cycle e pari a 1 nsec per la micro-architettura multi-cycle e considerando il codice in figura, quale delle seguenti affermazioni è vera?

```
loop:  LDR R0, [R1], #4
      ADD R2, R2, R0
      ADD R3, R3, #1
      CMP R3, R4
      BNE loop
```

Contrassegna solo un ovale.

- L'esecuzione di una iterazione sul single-cycle richiede 25 nsec e sul multi-cycle 5 nsec
- L'esecuzione di una iterazione sul single-cycle richiede 25 nsec e sul multi-cycle 20 nsec
- L'esecuzione di una iterazione sul single-cycle richiede 25 nsec e sul multi-cycle 15 nsec
- Nessuna delle precedenti

16. Si supponga di eseguire su un processore pipeline un'iterazione del ciclo riportato in figura. Qual'è il valore CPI corretto?

```
loop:  LDR R0, [R1], #4
      ADD R2, R2, R0
      ADD R3, R3, #1
      CMP R3, R4
      BNE loop
```

Contrassegna solo un ovale.

- 10/5
- 9/5
- 8/5
- 7/5
- 6/5
- 5/5

17. Quale componente di una memoria modulare interallacciata con K moduli va eliminato qualora si vogliono leggere K valori consecutivi di memoria a partire dall'indirizzo 0x00000001?

Contrassegna solo un ovale.

- Il multiplexer che sceglie fra le uscite dei K moduli indirizzati con i bit meno significativi dell'indirizzo quello giusto (ovvero quello indicato dai bit più significativi dell'indirizzo)
- Il demultiplexer che invia il segnale di WE al solo modulo che deve scrivere la parola di indirizzo dato fra i K moduli della memoria modulare
- Nessuno dei due casi precedenti

18. Il bit "I" in una istruzione operativa

Contrassegna solo un ovale.

- Permette di interpretare correttamente il contenuto del campo SRC2
- Permette di interpretare il campo RS come costante "piccola" (valore compreso fra 0 e 15) anziché come indice di registro
- Permette di indicare alla micro-architettura la necessità di scrivere i flag nelle istruzioni operative diverse dalla CMP
- Permette di indicare alla micro-architettura la necessità di scrivere i flag nelle istruzioni di salto condizionale

19. Nel codice in figura, eseguito su un processore con micro-architettura pipeline

```
i1:    add r1, r1, r2
i2:    sub r3, r3, r1
i3:    ldr r4, [r3, r6]
i4:    add r5, r5, r4
i5:    add r6, r6, #1
```

Contrassegna solo un ovale.

- Possiamo spostare l'istruzione i5 fra la i2 e la i3 per ridurre l'effetto della dipendenza fra la sub e la ldr che altrimenti richiederebbe un ciclo di stallo
- Possiamo spostare la i5 fra la i3 e la i4 per ridurre l'effetto della dipendenza fra la ldr e la add che altrimenti richiederebbe un ciclo di stallo
- Non c'è nessuna dipendenza logica che provochi stalli, e dunque non è necessario né opportuno riordinare le istruzioni al fine di ottenere tempi di esecuzione più brevi di quelli verificati nell'esecuzione del codice originale
- Il numero delle dipendenze logiche presenti e il tipo di istruzioni nella sequenza non permettono alcuna ristrutturazione delle istruzioni (riordino)

20. In quale micro-architettura non è necessario l'utilizzo di ALU dedicate per la gestione del registro PC, ma possiamo utilizzare l'ALU principale per calcolare l'indirizzo della prossima istruzione da eseguire?

Contrassegna solo un ovale.

- Nella micro-architettura single-cycle: il calcolo dell'indirizzo dell'istruzione seguente sia nel caso di una istruzione di salto che di tipo operativo e memora viene calcolato utilizzando la ALU principale utilizzata per calcolare le normali operative (tipo la ADD Rx, Ry, Rz)
- Nella micrarchitettura multi-cycle
- Nell'architettura pipeline
- Nessuna delle precedenti

21. Quanti cicli di clock servono per completare il seguente frammento di codice su un processore multi-cycle? Si assuma che il salto a "fine" non sia preso, quindi la sequenza comprende le 10 istruzioni dalla STR alla BNE (indicare nella risposta SOLO il numero dei cicli) Risposta: 40

```
main:   str r4, [sp], #-4      @ push r4
        mov r4, #0
        cmp r0, #0
        beq fine
        ldr r1, [r0]
        ldr r2, [r0, #4]
        add r1, r1, r2
        str r1, [r0], #8
        cmp r4, r3
        bne loop
```

22. Si considerino le istruzioni in figura. Quale delle sequenze soddisfa le condizioni di Bernstein e pertanto può essere considerata equivalente alla sequenza in figura?

```
add r1, r2, r3
mov r4, #0
sub r2, r5, r6
mul r7, r5, r2
```

Contrassegna solo un ovale.

- sub, add, mov, mul
- add, mov, mul, sub
- mov, add, sub, mul
- Nessuna delle precedenti

Questi contenuti non sono creati né avallati da Google.

Google Moduli

Quarta prova intermedia di AESO B

* Indica una domanda obbligatoria

1. Email *

2. Numero di matricola *

3. Quale delle seguenti affermazioni descrive meglio l'utilità del Write Buffer in una cache con politica delle scritture di tipo Write-Back?

Contrassegna solo un ovale.

- Il Write Buffer garantisce la coerenza dei dati tra cache differenti in un sistema multiprocessore.
- Il Write Buffer riduce la latenza di lettura dei dati in cache da parte della CPU
- Il Write Buffer viene utilizzato esclusivamente per ridurre i tempi di scrittura in memoria in caso in cui una linea di cache venga espulsa
- Il Write Buffer memorizza temporaneamente i dati da scrivere nella gerarchia di memoria
- Nessuna delle affermazioni precedenti è corretta

4. Si consideri una cache set associativa a 4 vie di capacità 128 KB, con linee di cache (blocchi) da 64 byte e parole da 32 bit. Quanti bit vengono usati per il tag, l'indice di blocco e l'offset di blocco (TAG, IDX, OFF)?

Contrassegna solo un ovale.

(15, 11, 6)

(18, 9, 5)

(17, 9, 6)

(17, 11, 4)

Nessuna delle precedenti

$2^7 * 2^{10} / 2^6 = 2^{11}$ linee di cache in totale
 I set sono $2^{11} / 4 = 2^9$ -> 9 bit per IDX
 2 bit per il byte-offset
 4 bit per il block-offset -> 6 bit per OFF
 $32 - 9 - 6 = 17$ bit per il TAG

5. In una gerarchia di memoria a tre livelli, la cache L1 ha un tempo di accesso di 1 ciclo, la cache L2 un tempo di accesso di 10 cicli, e la memoria un tempo di accesso di 100 cicli. Il Global Miss Rate L1-L2 è pari 0.2%. I cicli di stallo della CPU per gli accessi in memoria sono 2,2. Determinare il miss rate locale di L1 ed il miss rate locale di L2.

Si ricorda che il Global Miss Rate L1-L2 è la probabilità complessiva che un riferimento non venga soddisfatto né da L1 né da L2 ($\text{MissRateL1} * \text{MissRateL2}$).

Contrassegna solo un ovale.

Miss Rate L1 = 20%, Miss Rate L2 = 1%

Miss Rate L1 = 2%, Miss Rate L2 = 10%

Miss Rate L1 = 0,2%, Miss Rate L2 = 0,01%

Non si può calcolare perché dobbiamo conoscere il CPI perfetto (senza stalli)

$\text{CPI}_{\text{stall}} = \text{MRL1} * \text{MPL1} + \text{GMRL1L2} * \text{MPL2}$
 $2,2 = \text{MRL1} * 10 + 0,002 * 100$
 $\text{MRL1} = 0,2$ -> 20%
 $\text{GMRL1L2} = \text{MRL1} * \text{MRL2}$
 $\text{MRL2} = 0,002 / 0,2 = 0,01$ -> 1%

6. Si consideri una cache ad accesso diretto con 256 blocchi (linee di cache) ed 8 parole per blocco. Una parola è lunga 64 bit. Dato l'indirizzo esadecimale 0xFEAC, determinare l'indice di blocco ed il block offset della linea di cache che contiene la parola Mem[0xFEAC].

0xFEAC = 0...0 1111 1110 1010 1100
 byte-offset 3 bit (100)
 block-offset 3 bit (101 = 5)
 index 8 bit (11111010 = 250)

Contrassegna solo un ovale.

Indice di blocco 117, offset nel blocco 3 (cioè 118° linea, 4° parola)

Indice di blocco 250, offset nel blocco 5 (cioè 251° linea, 6° parola)

Indice di blocco 255, offset nel blocco 7 (cioè 255° linea, 8° parola)

7. Si consideri il seguente snippet di codice C, dove foo è una funzione che non modifica la località degli accessi. Il codice viene eseguito su un sistema che ha una cache dati di capacità $2N$ parole. Stimare il numero di fault di cache a regime fornendo un ordine di grandezza.

```
for (int i = 0; i < N; ++i)
  for (int j = 0; j < N; ++j)
    C[i][j] = foo( D[ (i + j) % N] );
```

Contrassegna solo un ovale.

$O(b)$

$O(N)$

$O(N/b)$

$O(b^2)$

$O(N^2 / b)$

A regime non ci sono fault di cache

C e D esibiscono solo località spaziale
quindi avremo N/b fault per D
e $N*N/b$ fault per C
Quindi l'ordine di grandezza è $O(N^2/b)$

8. Si consideri un sistema con indirizzi a 32 bit, cache dati ad accesso diretto di capacità $C=64$ parole e con blocchi (linee) da 32 byte. Supponendo che la cache sia inizialmente vuota, e che l'algoritmo di rimpiazzamento è LRU, qual è il miss rate (per la sola cache dati) generato dal frammento di codice C seguente? Effettuare il troncamento alla quarta cifra decimale.

```
for (int i = 1; i < 128; ++i)
  A[i] = A[i-1];
```

Contrassegna solo un ovale.

6,00%

3,00%

3,12%

6,29%

6,20%

3,14%

MR = #fault / (#totale di accessi)
#totale di accessi =
127 letture ($A[0] \dots A[126]$) +
127 scritture ($A[1] \dots A[127]$) = 254
#fault = $128/8 = 16$
MR = $16/254 = 0,0629 \rightarrow 6,29\%$

9. Un dispositivo trasferisce dati in modo continuo a una velocità di 16000 byte/s e genera una interruzione quando il buffer di I/O è pieno. La gestione dell'interruzione richiede 100 microsecondi di overhead fisso più 5 microsecondi per ogni byte da trasferire dal buffer. Supponendo che il device abbia 2 buffer di 16byte, quale frazione del tempo del processore viene consumato dalla gestione del dispositivo di I/O?

Contrassegna solo un ovale.

- 12%
- 15%
- 16%
- 18%

Viene generata una interruzione ogni
 $16000 \text{ byte/s} / 16 \text{ byte/inter} = 1000 \text{ inter/s}$
Ogni interruzione mi costa:
 $100 + 5 * 16 = 180 \text{ microsecondi}$
In totale spendo 180000 microsecondi (0,18s)
Per ogni secondo di esecuzione il processore
spende 0,18s per la gestione delle interruzioni
cioè il 18% del suo tempo

10. Il meccanismo del polling per la gestione di un dispositivo di I/O è una opzione valida se:

Contrassegna solo un ovale.

- il dispositivo ha una bassa banda di trasmissione
- il dispositivo ha una alta banda di trasmissione
- il dispositivo genera molte interruzioni nell'unità di tempo

11. Quali delle seguenti affermazioni è FALSA

Contrassegna solo un ovale.

- Per dispositivi ad alta banda, il meccanismo di gestione migliore è interrupt-driven I/O + DMA
- Il DMA controller di un dispositivo di I/O ad alta banda può avere precedenza rispetto alla CPU nell'accesso alla memoria principale
- L'indirizzo che la CPU fornisce al DMA controller è sempre un indirizzo virtuale.
- L'accesso all'area di memoria che mappa i registri di un dispositivo di I/O è accessibile solo in modalità "kernel"

12. Quali tra le seguenti funzionalità NON è da considerare un supporto hardware/firmware per il Sistema Operativo:

Contrassegna solo un ovale.

- la gestione dei livelli di privilegio
- la presenza di un dispositivo timer
- la protezione della memoria
- il meccanismo delle upcall
- la presenza di istruzioni privilegiate
- il mascheramento delle interruzioni

13. Quale delle seguenti affermazioni sui registri banked dell'architettura ARM è corretta?

Contrassegna solo un ovale.

- I registri banked sono disponibili in tutti i livelli di privilegio e vengono condivisi tra tutte le modalità operative
- I registri banked sono utilizzati solo per la gestione delle Fast Interrupt (FIQ)
- I registri banked sono i seguenti SPSR_svc, SPSR_irq, SPSR_fiq, SPSR_abt, SPSR_und e sono accessibili solo in modo kernel
- Nessuna delle precedenti affermazioni è corretta

14. Quale delle seguenti operazioni NON fa parte dell'implementazione del passaggio da stato user a stato kernel nel processore ARM ?

Contrassegna solo un ovale.

- Scrittura della CPSR
- Salvataggio dei registri R0-R7 sullo stack del kernel
- Attivazione dei banked register
- Salvataggio nel LR_banked del valore corrente del PC (user)
- Assegnazione al PC di un indirizzo nella parte bassa della memoria (0x000000XY)

15. Le istruzioni privilegiate:

Contrassegna solo un ovale.

- Si possono eseguire esclusivamente durante l'esecuzione di una syscall in modalità kernel
- Comprendono tutte e sole le istruzioni che servono per il trattamento delle interruzioni
- Comprendono tutte e sole le istruzioni che servono per gestire i dispositivi di I/O
- Nessuna delle precedenti affermazioni è vera

16. Qual è la differenza fra eccezioni e interruzioni?

Contrassegna solo un ovale.

- Le interruzioni hanno a che fare con la gestione dell'I/O mentre le eccezioni hanno esclusivamente a che fare con la gestione della memoria
- Le eccezioni hanno a che fare con la gestione dell'I/O mentre le interruzioni hanno esclusivamente a che fare con la gestione della memoria
- Le interruzioni sono eventi asincroni e le eccezioni sono eventi sincroni (rispetto all'esecuzione delle istruzioni sul processore)
- Le interruzioni sono eventi sincroni e le eccezioni sono eventi asincroni (rispetto all'esecuzione delle istruzioni sul processore)
- Non c'è differenza tra eccezioni ed interruzioni.

17. Quale struttura dati contiene, in ognuno dei suoi elementi, informazioni specifiche di una particolare unità di ingresso/uscita?

Contrassegna solo un ovale.

- Exception vector table, che in ARM si trova dall'indirizzo 0x00000000 all'indirizzo 0x00000024 (parte bassa della memoria)
- L'interrupt vector table
- La tabella dei descrittori di processo (PCB), limitatamente ai processi in esecuzione o nella lista dei processi pronti
- Nessuna delle precedenti

18. Quale dei seguenti arbitri ha linee dedicate per ognuno dei dispositivi che sottopone all'arbitraggio?

Contrassegna solo un ovale.

- Arbitro daisy chaining
- Arbitro a richieste indipendenti
- Qualunque tipo di arbitro che controlli gli accessi in memoria
- L'arbitro che controlla gli accessi alle cache di primo livello

19. Il motivo del maggior numero di registri banked nelle Fast Interrupt (FIQ) è:

Contrassegna solo un ovale.

- Mettere a disposizione un numero di registri tale per cui le interruzioni più "semplici" da trattare non necessitino di accessi allo stack utente per il salvataggio dei registri
- Mettere a disposizione un numero di registri tale per cui le interruzioni più "semplici" da trattare non necessitino di accessi allo stack kernel per il salvataggio dei registri
- Mettere a disposizione un numero di registri tale per cui tutte le interruzioni non necessitino di accessi allo stack kernel per il salvataggio dei registri
- Mettere a disposizione un numero di registri tale per cui tutte le interruzioni non necessitino di accessi allo stack utente per il salvataggio dei registri
- Semplificare il processo di trasferimento della parola che rappresenta il motivo (quale dispositivo e perché) dell'interruzione nella prima fase del trattamento dell'interruzione

20. Durante l'esecuzione di una system call (SVC/SWI) in ARM:

Contrassegna solo un ovale.

- Si disabilitano solo le interruzioni Fast (FIQ)
- Si disabilitano sia le interruzioni Fast (FIQ) che quelle normali (IRQ)
- Si disabilitano solo le interruzioni normali (IRQ)
- Non si disabilitano né le interruzioni Fast (FIQ) né quelle normali (IRQ)

21. Durante una upcall, quale di queste informazioni viene salvata sullo stack, prima di passare il controllo alla callback (upcall_handler) indicata dal processo utente?

Contrassegna solo un ovale.

- I registri che rappresentano lo stato interno del kernel al momento dell'evento che general la upcall
- La copia dei registri banked della modalità kernel SVC al momento dell'evento che general la upcall
- Il program counter del processo utente al momento dell'evento che general la upcall unitamente ad altre informazioni
- La posizione del vettore delle interruzioni relativa al dispositivo che ha generato la upcall (per esempio il timer) in modo che la callback abbia accesso al codice dell'handler in modalità utente

22. In un sistema con cache, scaricare qualcosa dalla cache per far posto a qualcos'altro di utilizzo più immediato anche se ciò che si scarica fa parte del working set:

Contrassegna solo un ovale.

- Avviene solo nel caso di utilizzo di cache a indirizzamento diretto
- Avviene solo nel caso di utilizzo di cache a indirizzamento set associativo
- Avviene solo nel caso di utilizzo di cache a indirizzamento completamente associativo
- Avviene solo quando si utilizzino cache ad accesso diretto o completamente associativo
- Può avvenire con cache che utilizzino uno qualunque dei modi di indirizzamento

23. Nel caso in cui la cache di primo livello (dati e istruzioni) sia più piccola della dimensione del working set di un certo programma in esecuzione:

Contrassegna solo un ovale.

- Potremmo avere un numero di fault sicuramente maggiore del numero di fault fisiologici (compulsory)
- Avremo un numero di fault sicuramente maggiore del numero di fault fisiologici (compulsory)
- Possiamo comunque trasformare il codice in modo da avere un numero di fault pari alla dimensione del codice e dei dati divisa per la dimensione della linee della cache, qualunque sia il modo di indirizzamento

24. Quale delle seguenti affermazioni è corretta?

Contrassegna solo un ovale.

- Il fault compulsory (fisiologici) si verifica con qualunque tipo di cache
- Il fault capacity si verifica solo con cache ad indirizzamento diretto
- Entrambi i tipi di fault si verificano solo con cache ad indirizzamento diretto o associativo su insiemi, mentre nelle cache completamente associative non si hanno fault capacity

25. Il Direct Memory Access (DMA)

Contrassegna solo un ovale.

- è utilizzato da tutti i dispositivi di I/O per velocizzare i trasferimenti da e per la memoria principale
- è utilizzato da tutti i dispositivi di I/O per velocizzare i trasferimenti da e per la memoria cache di secondo livello
- è utilizzato solo dai dispositivi "a blocchi" (es. dischi) che trasferiscono blocchi di byte per velocizzare i trasferimenti da e per la memoria principale
- è utilizzato dai dispositivi di I/O che trasferiscono blocchi di byte per non far gravare sul processore il peso dei trasferimenti fra memoria principale e memoria del dispositivo

Google Moduli

Quinta prova AESO 2024-25

* Indica una domanda obbligatoria

1. Email *

2. Numero di matricola *

3. Dato il codice in figura, quale delle seguenti affermazioni è VERA?

```
int main(int argc, char * argv) {
    int x = 123, status;
    int fd[2];
    pipe(fd);

    int pid = fork();
    if(pid) {
        close(fd[0]);
        write(fd[1], &x, sizeof(int));
        close(fd[1]);
        wait(&status);
    } else {
        int m;
        while(read(fd[0], &m, sizeof(int)) != 0)
            printf("Ricevuto %d\n", m);
        close(fd[0]);
    }
    return(0);
}
```

Contrassegna solo un ovale.

- Termina senza stampare nulla
- Termina dopo aver stampato "Ricevuto 123"
- Termina dopo aver stampato "Ricevuto 123" e "Ricevuto -1"
- Non termina perchè non si controllano mai i valori di ritorno delle chiamate di sistema (pipe, fork, read, write e close)
- Non termina perché il processo figlio rimane bloccato sulla read
- Nessuna delle precedenti affermazioni è vera

Rimane bloccato sulla read perché il processo figlio non ha chiuso il descrittore di scrittura della pipe (fd[1]) e di conseguenza non riceve l'End-Of-File (EOF) che sblocca la read facendole ritornare 0.

4. Quale delle seguenti affermazioni, relative ad una chiamata di sistema "read" è FALSA?

Contrassegna solo un ovale.

- restituisce un intero maggiore o uguale a zero in caso di terminazione con successo, o -1 in caso di errore
- se restituisce un numero maggiore di 0, esso rappresenta il numero di caratteri effettivamente letti
- restituisce sempre un valore diverso da zero
- in certi casi può restituire il valore 0

5. Quale delle seguenti informazioni NON appartiene alle informazioni contenute nel PCB (process control block) se si considera l'architettura ARMv7?

Contrassegna solo un ovale.

- L'indirizzo di ritorno della procedura corrente
- Il contenuto dei registri generali
- Lo stato interno relativo alle variabili utilizzate per la sincronizzazione del processo (mutex, variabili condizione o semafori)
- Il valore della parola di stato
- Il valore del PC

6. In una syscall:

Contrassegna solo un ovale.

- Il controllo sulla correttezza dei parametri viene effettuato nella stub user mode, prima di passare in modalità kernel
- Il controllo sulla correttezza dei parametri viene effettuato sempre ed esclusivamente una volta passati in modalità kernel
- Il controllo dei parametri viene effettuato in parte nella stub in user mode e in parte in modalità kernel
- non viene effettuato alcun controllo, dal momento che qualunque valore dei registri utilizzati per i parametri è legale. In caso, l'esecuzione della chiamata di sistema ritornerà un errore.

7. Quando si effettua una `acquire()` su una mutex:

Contrassegna solo un ovale.

- il processo o thread chiamante può finire in stato di wait, in caso la mutex sia stata già oggetto di una `acquire()` senza che successivamente sia stata eseguita una `release()`
- il processo o thread che esegue la `acquire()` non finisce mai in stato di attesa. Se qualche altro processo o thread ha già effettuato una `acquire()` sul mutex, senza successivamente (ancora) aver eseguito una `release()`, al chiamante viene restituito un booleano FALSE
- il processo o thread che esegue la `acquire()` può finire in stato di attesa o no, a seconda del parametro della `acquire` che indica che il mutex è un mutex normale o uno spinlock

8. Si considerino le seguenti coppie di strutture dati. Quante delle coppie sono tali per cui la prima struttura dati deve essere utilizzata necessariamente insieme ad una istanza della seconda?

Variable condizione - Spinlock	NO
Variable condizione - Mutex	SI
Spinlock - Variable condizione	NO
Mutex - Variable condizione	NO
Mutex - Spinlock	NO
Spinlock - Mutex	NO
Semaforo - Mutex	NO
Mutex - Semaforo	NO

Contrassegna solo un ovale.

- Nessuna
- 1
- 2
- 3
- 4
- 5
- 6
- 7
- Tutte

Le variabili di condizione necessitano di una mutex, altrimenti non possono essere usate.

9. Quale delle seguenti condizioni NON è una delle condizioni che indicano la possibilità di raggiungere uno stato di stallo?

Contrassegna solo un ovale.

- Risorse limitate
- Hold while waiting (o Wait while holding)
- Accesso a risorse in ordine diverso da parte di thread diversi
- Attesa circolare
- Assenza della possibilità di prerilascio

10. La starvation

Contrassegna solo un ovale.

- è la condizione in cui un thread non procede nella propria esecuzione perchè richiede una risorsa assegnata ad un secondo thread che a sua volta richiede l'accesso ad una risorsa già assegnata a questo thread
- è la condizione in cui ogni volta che un thread ottiene un quanto di tempo per l'utilizzo del processore subisce un prerilascio a causa di un thread a più alta priorità e quindi non completa mai il suo quanto di tempo
- è la condizione in cui un thread non procede nella propria esecuzione, in assenza di situazioni di stallo, con altri thread che riescono a ottenere l'utilizzo del processore senza permettergliene l'utilizzo

11. Quale di queste tecniche relative allo stallo può essere classificata come detect and fix?

Contrassegna solo un ovale.

- Algoritmo del banchiere
- Utilizzo di spinlock invece dei normali mutex bloccanti
- Analisi delle attività concorrenti in attesa di risorse, riconoscimento di una situazione di attesa circolare e conseguenze terminazione di uno dei thread coinvolti nell'attesa circolare
- Rimozione della condizione "wait while holding"

12. L' algoritmo del banchiere

Contrassegna solo un ovale.

- va eseguito ogni volta che si rilascia una risorsa
- va eseguito ogni volta che si richiede una risorsa
- va eseguito sia quando si richiede che quando si rilascia una risorsa

13. La figura riporta la molteplicità delle risorse, l' assegnazione corrente e l' esigenza residua.

Il thread A richiede una risorsa di R1, successivamente il thread C richiede una risorsa di R3. Secondo l' algoritmo del banchiere, quali richieste possiamo accettare?

Molteplicità			
R1	R2	R3	
4	4	4	

Assegnamento			
R1	R2	R3	
A	2	3	0
B	0	0	2
C	0	0	1

Esigenza			
R1	R2	R3	
A	2	0	1
B	2	1	0
C	0	0	2

Disponibilità prima della richiesta di A

R1	R2	R3
2	1	1

Situazione dopo l' ipotetico assegnamento di R1 ad A

Disponibilità

R1	R2	R3
1	1	1

Assegnamento

	R1	R2	R3
A	3	3	0
B	0	0	2
C	0	0	1

Esigenza

	R1	R2	R3
A	1	0	1
B	2	1	0
C	0	0	2

Lo stato è sicuro perché:

A può terminare, Disponibilità (4,4,1)

B può terminare, Disponibilità (4,4,2)

C può terminare, Disponibilità (4,4,4)

Quindi la richiesta di A viene soddisfatta.

Dal nuovo stato sicuro valutiamo se anche la richiesta di C di una risorsa R3 può essere soddisfatta.

Stato dopo l' ipotetico assegnamento di R3 a B

Disponibilità

R1	R2	R3
1	1	0

Esigenza

	R1	R2	R3
A	1	0	1
B	2	1	0
C	0	0	1

Assegnamento

	R1	R2	R3
A	3	3	0
B	0	0	2
C	0	0	2

La disponibilità è insufficiente per soddisfare le esigenze dei 3 processi. Lo stato non è sicuro e la richiesta di C non può essere soddisfatta.

Contrassegna solo un ovale.

- Entrambe le richieste vengono soddisfatte
- Nessuna delle due richieste vengono soddisfatte, entrambi i thread rimangono in attesa
- Solo la richiesta di C viene soddisfatta, il thread A rimane in attesa.
- Solo la richiesta di A viene soddisfatta, il thread C rimane in attesa.
- Lo stato di partenza non è uno stato sicuro e quindi non è possibile rispondere

14. Quale di queste istruzioni, in nessuna circostanza causa un cambiamento di stato (per esempio da utente a kernel o viceversa)?

Contrassegna solo un ovale.

- SVC
 SWI
 BL
 MOVS

15. In una upcall, sullo stack utente disponibile al momento in cui viene eseguita la funzione callback (upcall handler):

Contrassegna solo un ovale.

- E' presente lo stato architetturale al momento in cui si è verificato l'evento che ha generato l'upcall, ma solo in modalità lettura
 E' presente lo stato architetturale al momento in cui si è verificato l'evento che ha generato l'upcall, sia in modalità lettura che in modalità scrittura
 Non è disponibile in alcun modo lo stato architetturale al momento in cui si è verificato l'evento che ha causato l'upcall

16. Due thread intendono entrambi utilizzare una risorsa di tipo A e una di tipo B per poter completare la loro esecuzione. Quali delle seguenti ipotesi evita la possibilità di avere stallo?

Contrassegna solo un ovale.

- Thread1: lockA.acquire(); lockB.acquire(); calcolo; lockA.release(); lockB.release(); Thread2: lockB.acquire(); lockA.acquire(); calcolo; lockB.release(); lockA.release();
 Thread1: lockA.acquire(); lockB.acquire(); calcolo; lockA.release(); lockB.release(); Thread2: lockA.acquire(); lockB.acquire(); calcolo; lockB.release(); lockA.release();
 Thread1: lockA.acquire(); spinlockB.trylock(); calcolo; lockA.release(); lockB.release(); Thread2: spinlockB.trylock(); lockA.acquire(); calcolo; lockB.release(); lockA.release();
 Nessuna delle sequenze precedenti evita lo stallo.

17. Quale delle seguenti politiche minimizza il tempo di permanenza dei processi nel sistema (turnaround time)?

Contrassegna solo un ovale.

- FIFO
- SJF
- ROUND ROBIN con PRIORITA'
- ROUND ROBIN con PRERILASCIO

18. Dati i thread in figura, dire il tempo in cui termina il thread B, assunto uno scheduler round robin con quanti di tempo di 10ms, che non usa priorità né pre-rilascio.

Thr	Arrivo	Durata
A	t=0	50ms
B	t=2	20ms
C	t=5	10Ms

Contrassegna solo un ovale.

- 20ms
- 30ms
- 40ms
- 50ms
- 60ms
- 70ms
- 80ms

19. Uno scheduler MFQ ha 4 code con priorità decrescente e lavora con quanti di tempo. I processi che consumano l'intero quanto di tempo, perdono priorità (priorità precedente - 1). Quelli che rilasciano il processore prima dello scadere del quanto di tempo acquistano priorità (priorità precedente + 1). Ci sono 4 processi (A, B, C e D). A e B hanno priorità 4 e consumano sempre tutti i loro quanti di tempo. C e D hanno priorità 3 e 1, rispettivamente, e rilasciano sempre il processore dopo esattamente metà quanto di tempo facendo una `thread_yield()`. Il processo che rilascia il processore viene inserito all'ultimo posto della coda dei processi pronti relativa alla sua priorità. Quale situazione abbiamo dopo un tempo pari a 3 quanti di tempo a partire dalla situazione delle code riportata in figura, supponendo che tutti i processi necessitino ancora di eseguire per 3 interi quanti di tempo, eventualmente frazionati in più periodi di lunghezza minore ?

Priorità 4	RDY : A -> B
Priorità 3	RDY : C
Priorità 2	RDY :
Priorità 1	RDY : D

Contrassegna solo un ovale.

- E' in esecuzione il processo A
- E' in esecuzione il processo B
- E' in esecuzione il processo C
- E' in esecuzione il processo D

Parte A che esegue per 1 QdT e al termine va in Coda3
Parte B che esegue per 1 QdT e al termina va in Coda3
Parte C che esegue per 1/2 QdT ed al termine va in Coda4
Parte nuovamente C che esegue per 1/2 QdT e rimane in Coda4

20. Si consideri il codice che segue, relativo ad un processo consumatore che accede una mailbox di messaggi. Quale delle seguenti affermazioni riflette correttamente il comportamento desiderato?

```
mutex.acquire();
while(mailbox.empty()) emptyCV.wait(&mutex);
mutex.release();
mailbox.getMessage();
fullCV.signal();
```

Contrassegna solo un ovale.

- la signal dovrebbe avere come parametro il mutex NO con semantica MESA
- la mutex.release() dovrebbe essere fatta dopo la getMessage()
- la signal dovrebbe essere fatta prima della getMessage()
- la signal dovrebbe essere fatta prima della mutex.release() NO, in entrambi i casi la getMessage() rimarrebbe fuori dalla sezione critica.

21. Si consideri il problema dei filosofi a cena, in cui sono presenti 4 filosofi e 4 bacchette/forchette. L'implementazione che prevede l'ordinamento sull'indice dei filosofi:

Contrassegna solo un ovale.

- Non evita il deadlock dei filosofi
- Evita il deadlock e permette sempre ad almeno due filosofi di mangiare concorrentemente
- Evita il deadlock ma in alcuni casi è possibile che un solo filosofo mangi mentre gli altri sono bloccati
- Evita la starvation e non il deadlock
- Evita la starvation ed il deadlock

L'implementazione prevede che i filosofi di indice pari prendano prima la bacchetta i e poi la bacchetta $i+1 \text{ mod } 4$, i filosofi di indice dispari prima la bacchetta $i+1 \text{ mod } 4$ e poi la bacchetta i . Questo algoritmo rompe la condizione di circular waiting ed è quindi immune da deadlock. Non è però immune da starvation. Non è garantito sempre che due filosofi riescano a mangiare, dipende dallo scheduling dei thread filosofi.

22. Si consideri il codice nella figura che implementa i metodi put e get di un buffer bounded multi-producer/multi-consumer, utilizzando un'unica variabile condizione (cv). Quale delle seguenti affermazioni è vera:

```

void put(X) {
    lock.acquire();
    while(buffer.full()) {
        cv.wait(&lock);
    }
    insert(buffer, X);
    cv.signal();
    lock.release();
}

void get(&X) {
    lock.acquire();
    while(buffer.empty()) {
        cv.wait(&lock);
    }
    extract(buffer, &X);
    cv.broadcast();
    lock.release();
}

```

Contrassegna solo un ovale.

- Il programma è corretto anche se l'uso della cv.broadcast() è sconsigliato
- Il programma è corretto anche se inefficiente dato che utilizza una sola variabile di condizione invece di due.
- Il programma non è corretto perché la cv.broadcast() dovrebbe essere eseguita dopo la lock.release()
- Il programma non è corretto perché invece di cv.signal() nella put dovrebbe usare cv.broadcast()

- Nessuna delle precedenti è vera

Il programma così come scritto non è corretto perché almeno nel caso di buffer di capacità 1 con N produttori ed 1 consumatore, la signal nella put non garantisce il risveglio del consumatore con conseguente rischio di deadlock.

23. Si consideri l'algorithmo di scheduling per multiprocessori di tipo Space Sharing. Quale delle seguenti affermazioni è VERA?

Contrassegna solo un ovale.

- Lo scheduling consente a più processi di condividere contemporaneamente lo stesso insieme di core o nodi
- Lo scheduling prevede di dare priorità ai job sequenziali
- Lo scheduling assegna l'uso esclusivo di un set di CPU ad un singolo job finché non termina
- Lo scheduling dà priorità ai job paralleli che necessitano di tutte le CPU

24. Si consideri un sistema con scheduling Round Robin con quanto di tempo di 10ms e gestione con pririlascio. Nel sistema sono presenti i thread A, B e C. Al tempo t , il thread A è appena passato in esecuzione, B è in attesa sul semaforo Sem e C è in stato di pronto e detiene la spinlock SL. Determinare quale thread è in esecuzione al tempo $t+20$ se si verificano in sequenza i seguenti eventi:
- $t+5$ il thread in esecuzione esegue $V(\text{sem})$
 - $t+7$ il thread in esecuzione esegue $P(\text{sem})$
 - $t+10$ il thread in esecuzione esegue $\text{spinlockRelease}(\text{SL})$
 - $t+20$ il thread in esecuzione esegue $P(\text{sem})$

Contrassegna solo un ovale.

	t	Exec	Ready	Sem	SL
<input type="radio"/> Il thread A	0	A	C	0, B	BUSY
<input type="radio"/> il thread B	5	A	C->B	0, -	BUSY
<input checked="" type="radio"/> il thread C	7	C	B	0, A	BUSY
<input type="radio"/> nessuno è in esecuzione perché si ha deadlock	10	C	B	0, A	READY
	17	B	C	0, A	READY
	20	C	-	0, A->B	READY

Questi contenuti non sono creati né avallati da Google.

Google Moduli

Sesta verifica AESO 2024-25

* Indica una domanda obbligatoria

1. Email *

2. Numero di matricola *

3. La inverted page table

Contrassegna solo un ovale.

- ha tante entry quante sono le pagine logiche in uso da parte dei processi in esecuzione
- ha tante entry quanto sono le pagine di memoria fisica
- ha tante entry quanto sono le pagine di memoria virtuale dei processi in esecuzione

4. Il metodo Base & Bound:

Contrassegna solo un ovale.

- Permette di ottenere l'indirizzo fisico dall'indirizzo logico e da una tabella, puntata dal registro BASE mediante giustapposizione di parte dei bit dell'indirizzo logico con quelli ottenuto nell'accesso alla tabella
- Non permette di condividere codice fra processi diversi
- Evita il problema della frammentazione esterna

5. La deframmentazione della memoria, eseguita in un regime di allocazione a partizioni variabili:

Contrassegna solo un ovale.

- Richiede un tempo proporzionale al numero di partizioni libere
- Richiede un tempo proporzionale alle dimensioni delle partizioni allocate che debbono essere riallocate in posizioni diverse
- Richiede la copia di tutta la memoria fisica

6. Si consideri un sistema con segmentazione pura. Quale delle seguenti affermazioni è vera?

Contrassegna solo un ovale.

- Il calcolo dell'indirizzo fisico richiede l'accesso a due tabelle: la tabella dei segmenti e la tabella delle pagine. Dalla prima si ottiene un numero di segmento fisico, dalla seconda il numero di pagina fisica e i due valori sono concatenati con l'offset di segmento per ottenere l'indirizzo fisico
- L'indirizzo fisico è ottenuto da quello logico sostituendo il numero di segmento logico, nella parte alta dell'indirizzo logico, con il numero di indirizzo fisico ottenuto accedendo alla tabella dei segmenti del processo
- L'indirizzo fisico si ottiene sommando l'offset di segmento presente nell'indirizzo logico al valore "Base" ottenuto accedendo alla tabella dei segmenti, e dopo aver controllato che l'offset non superi la lunghezza del segmento

7. La condivisione del codice di una libreria, in presenza di segmentazione con paginazione:

Contrassegna solo un ovale.

- Avviene facendo in modo che tutti i segmenti logici utilizzati nei vari processi per accedere alla libreria permettano di ottenere un puntatore alla stessa tabella dei segmenti, che contiene tutti i segmenti fisici occupati dalla libreria
- Avviene facendo in modo che tutti i segmenti logici utilizzati nei vari processi per accedere alla libreria permettano di ottenere un puntatore alla stessa tabella delle pagine, contenente tutte le pagine occupate dalla libreria
- Avviene utilizzando in tutti i processi lo stesso numero di segmento logico per indirizzare la libreria

8. La traduzione di un indirizzo logico in un indirizzo fisico, in un sistema con paginazione

Contrassegna solo un ovale.

- Richiede una tabella delle pagine e un addizionatore
- Richiede una tabella delle pagine, un addizionatore e un confrontatore
- Richiede una tabella delle pagine e la possibilità di giustapporre sequenze di bit di lunghezza fissa (e.g. k e $(32-k)$ bit con k fissato)
- Richiede una tabella delle pagine e la possibilità di giustapporre sequenze di bit di lunghezza variabile (e.g. $\langle k, 32-k \rangle$ bit per certe pagine e $\langle h, 32-h \rangle$ bit per altri tipi di pagine)

9. In un sistema con paginazione, la lunghezza delle pagine:

Contrassegna solo un ovale.

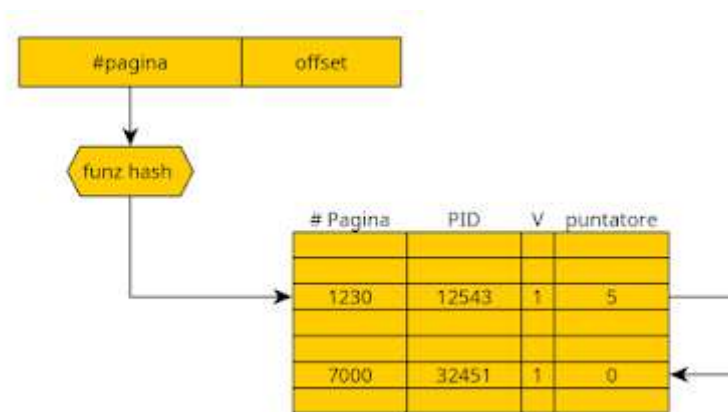
- E' decisa a livello del sistema operativo
- E' normalmente decisa dal sistema operativo e può essere modificata a tempo di esecuzione
- E' definita dall'architettura
- E' definta dall'architettura ma dipende comunque dal modo operativo (user o system mode)

10. Quale delle seguenti affermazione, relative alla dimensione della memoria fisica, è FALSA ?

Contrassegna solo un ovale.

- Ha un massimo che può essere derivato dalle strutture dati utilizzate per le traduzioni degli indirizzi
- non può eccedere 2^k , con k numero di bit dell'indirizzo fisico definito a livello architetturale
- dipende dal tipo di sistema operativo. In un sistema micro kernel è fissata dal numero di bit che l'architettura utilizza per gli indirizzi fisici. In un sistema operativo "monolitico" dipende dal tipo delle strutture dati utilizzate per la traduzione degli indirizzi
- Nessuna delle affermazioni precedenti è falsa

11. Per quale motivo nella "inverted page table" (core map) è presente un campo puntatore ?



Contrassegna solo un ovale.

- Per permettere l'accesso al numero di pagina fisica dal numero di pagina logica in presenza di una funzione hash che mappi numeri di pagina logica diversi sullo stesso numero di pagina fisica
- per permettere di accedere a tutte le pagine di un processo mediante una lista concatenata di numeri di pagine fisiche
- perchè ogni pagina fisica può contenere pagine logiche di processi diversi, in momenti diversi. La lista ottenuta utilizzando i puntatori permette di cercare la pagina logica corretta, una volta noto il PID del processo in esecuzione, seguendo la lista e confrontando il PID con il secondo campo della core map

12. In un sistema con paginazione, la TLB

Contrassegna solo un ovale.

- è una cache associativa a 2 vie, nella quale il tag è rappresentato dal numero di pagina fisica
- è una cache completamente associativa, con $b=4$ e tag composto dal numero di pagina logica e dal bit di presenza. Il fattore b "piccolo" è dovuto al fatto che la località spaziale negli accessi alla cache è relativamente modesta.
- è una cache completamente associativa, con $b=1$ e tag composto dal numero di pagina logica e dal bit di presenza
- è una cache completamente associativa, con $b=1$ e tag che corrisponde al numero di pagina logica

13. Si considerino Virtual e Physical cache. Quale delle seguenti affermazioni è FALSA?

Contrassegna solo un ovale.

- le cache differiscono per il tipo di indirizzamento, con indirizzi virtuali in un caso e fisici nell'altro
- le cache differiscono per il tipo di dati utilizzati per l'accesso: in un caso è presente il solo tag, nell'altro caso sia il tag che il pid
- le cache differiscono per il tipo di indirizzamento, diretto nelle cache fisiche e associativo su insiemi nelle cache virtuali
- cache di livelli diversi possono essere di tipo diverso (e.g. livello 1 -> virtuale, livello 2 -> fisica)

14. Il caricamento di una pagina in memoria centrale:

Contrassegna solo un ovale.

- Avviene quando si verifica un'interruzione, che provoca la messa in esecuzione di un processo che si trovava in attesa.
- Avviene in seguito ad un'eccezione "page fault", che riguarda uno dei processi nella lista dei pronti o il processo in esecuzione
- Avviene quando l'utente viene informato di un tentativo di accesso ad una pagina non presente in memoria centrale mediante una upcall fatta partire dall'eccezione page fault rilevata nella MMU
- Nessuna delle precedenti affermazioni è vera

15. I bit M (modifica) e U (uso) relativa a pagine residenti in memoria:

Contrassegna solo un ovale.

- Servono per la traduzione degli indirizzi logici in indirizzi fisici. Per esempio, U=0 indica che la pagina non è presente e quindi l'indirizzo fisico non può essere generato.
- Vengono utilizzati per decidere quale pagine selezionare come "vittima" a seguito di un page fault e in assenza di pagine libere in memoria centrale
- Servono per minimizzare la frammentazione esterna
- Servono per ottimizzare la gestione delle pagine del working set di processi diversi al momento della schedulazione

16. Si consideri il numero k di pagine riservate in memoria centrale per l'esecuzione di un generico processo, in un sistema operativo multi-user e multi-process.

Contrassegna solo un ovale.

- Più k è alto e migliori sono le prestazioni del sistema, visto il maggior numero di pagine a disposizione per l'esecuzione di ciascun processo
- Più k è limitato e migliori sono le prestazioni del sistema, dal momento che siamo in grado di allocare in memoria le pagine di più processi contemporaneamente
- il numero k va definito in modo che permetta sia di allocare più processi in memoria che di avere abbastanza pagine per contenere i working set dei processi in esecuzione
- k è un parametro che dipende esclusivamente dalla dimensione della memoria fisica

17. Si consideri il tempo scandito da una variabile di tipo intero, che viene incrementata ad ogni unità di tempo trascorsa. Nella politica di sostituzione MIN, quale pagina viene identificata come vittima?

Contrassegna solo un ovale.

- quella che ha il valore minimo nel campo "tempo dell'ultimo accesso"
- quella che ha il valore massimo nel campo "tempo del prossimo accesso"
- quella che ha il valore minimo nel campo "tempo dell'ultimo accesso", fra quelle che hanno il bit di modifica $M=0$
- nessuna delle precedenti

18. Utilizzando una politica FIFO per il rimpiazzamento delle pagine in memoria:

Contrassegna solo un ovale.

- La disponibilità di un numero maggiore di pagine fa diminuire il numero di fault di pagina, a parità di processi in esecuzione
- La disponibilità di un numero minore di pagine fa aumentare il numero di fault di pagina, a parità di processi in esecuzione
- Il numero di fault di pagina, per lo stesso insieme di processi, può decrescere in presenza di un minor numero di pagine disponibili in memoria
- Nessuna delle precedenti risposte è vera

19. Quale delle seguenti affermazioni, relative all'algorithmo second chance, è FALSA?

Contrassegna solo un ovale.

- pagine non riferite recentemente sono immediatamente selezionate come vittime
- pagine non riferite recentemente non sono immediatamente selezionate come vittime ma viene offerta una seconda opportunità per rimanere in memoria
- pagine riferite recentemente possono essere immediatamente selezionate come vittime, in condizioni particolari

20. Nella paginazione, il fenomeno del trashing:

Contrassegna solo un ovale.

- Fa diminuire il tempo di permanenza medio dei processi in esecuzione
- Può causare un degrado delle prestazioni anche di diversi ordini di grandezza
- Di solito è relativo a un piccolo sottoinsieme dei processi in esecuzione che utilizzano le stesse pagine fisiche per ospitare le proprie pagine logiche durante l'esecuzione

21. In un dispositivo SSD il Flash Translation Layer

Contrassegna solo un ovale.

- codifica una corrispondenza fra blocchi logici e blocchi fisici
- fa corrispondere a ciascun blocco logico un numero di settore e di traccia che ne permettono l'individuazione sul dispositivo
- è utilizzata esclusivamente per ottimizzare i tempi di scrittura

22. In un file system di tipo UNIX FFS i blocchi del disco hanno ampiezza di 2KByte e i puntatori ai blocchi sono a 32 bit. Gli i-node contengono, oltre agli altri attributi, 10 puntatori diretti e 3 puntatori indiretti (indiretto singolo, doppio e triplo). Quanti blocchi dati, al massimo, possono essere riferiti utilizzando il puntatore indiretto doppio?

Contrassegna solo un ovale.

- 2²⁰
- 2¹⁹
- 2¹⁸
- 2¹⁷
- 2¹⁶

$2^{11}/2^2 = 2^9$ n. di puntatori in ogni blocco indice
I blocchi dati riferiti dal puntatore indiretto doppio è $2^9 * 2^9 = 2^{18}$

23. In un file system di tipo UNIX FFS i blocchi del disco hanno ampiezza di 1Kbyte e gli i-node contengono 10 indirizzi diretti e 3 indirizzi indiretti (singolo, doppio e triplo). Il puntatore indiretto doppio punta ad un blocco indice di secondo livello che contiene puntatori a blocchi indice di primo livello i quali contengono puntatori a blocchi dati. Tutti gli indirizzi hanno una lunghezza di 32 bit. Un file ha dimensione 2MB. Indicare quanti puntatori a blocchi indiretti singoli sono utilizzati nel blocco indice di secondo livello.

Contrassegna solo un ovale.

- 6
- 7
- 8
- 9

Per il file servono 2048 blocchi (2MB/1KB)
Il numero di puntatori per blocco indice è 256 (1KB/4byte)
 $2048 - 10 - 256 = 1782$ numero di blocchi residui
 $1782 / 256 = 6.96$ quindi servono 7 blocchi indice di primo livello
indicizzati dal blocco indice di secondo livello

24. Quale delle seguenti affermazioni è vera?

Contrassegna solo un ovale.

- La FAT deve risiedere tutta in memoria centrale
- La tabella degli I-nodi di un file system FFS deve risiedere tutta in memoria principale
- La MFT di NTFS contiene un record per ciascuno dei file nel FS e solo i record dei file aperti sono mantenuti in memoria centrale

25. In un file system FFS, il minimo numero di accessi al disco (in assenza di caching) per l'accesso ad un file è:

Contrassegna solo un ovale.

- 1
- 2
- 3
- proporzionale alla lunghezza del file
- proporzionale alla lunghezza dell'i-nodo

26. Un sistema effettua la traduzione degli indirizzi con segmentazione paginata a 2 livelli. Le pagine sono di 2Kbyte, l'indirizzo logico è di 48 bit. Il sistema ha le tabelle delle pagine tutte della stessa dimensione. Il numero di tabelle delle pagine di secondo livello per ogni segmento è pari a 16384. Si chiede di indicare il numero massimo di segmenti che può avere un processo.

Contrassegna solo un ovale.

- 128
- 256
- 512
- 1024
- 2048
- nessuna delle precedenti
- 11 bit di offset
14 bit per indice di primo livello
14 bit per indice di secondo livello
 $48 - 11 - 28 = 9$ bit utilizzati per il numero di segmenti
 $2^9 = 512$ segmenti per processo

Questi contenuti non sono creati né avallati da Google.

Google Moduli

