

Pretest Practice

Niccolò Gabrielli

August 6, 2025

Contents

1	24-01-2025	2
2	08-01-2025	5

Domande di Reti Logiche – compito del 24/01/2025

C'è una sola risposta corretta per ogni domanda
Usare lo spazio bianco sul retro del foglio per appunti, se serve

	b7	b0
100	0x1C	
101	0x89	
102	0xA2	
103	0xC6	

```
MOV 100, %EAX
CMP $0, %AX
JL dopo
```

1) Dato il contenuto della memoria riportato in figura, il codice scritto sopra salta all'etichetta *dopo*.

- Sempre
- Mai
- Mancano informazioni per poter decidere
- Nessuna delle precedenti

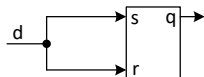
2) In complemento alla radice, la somma di due numeri interi *riducibili* è rappresentabile sul numero di cifre degli operandi

- In qualunque base
- In nessuna base
- Solo in base 2
- Nessuna delle precedenti

```
SAR %AL
ROL %AL
```

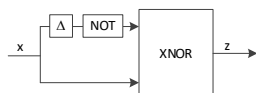
3) In AL c'è lo stesso contenuto *prima* e *dopo* il codice scritto sopra se:

- AL è minore di 128
- AL è multiplo di 2
- Solo se AL contiene un intero
- Nessuna delle precedenti



4) Si connetta *d* ad un clock di periodo *T* molto maggiore del tempo di risposta del latch SR. L'uscita *q*:

- È un clock di periodo *T*
- È un clock di periodo $2T$
- Cambia nel tempo in modo non prevedibile
- Nessuna delle precedenti



5) Nel circuito di sopra, l'uscita *z*

- Vale sempre uno
- Insegue l'ingresso con un ritardo circa Δ
- Genera un impulso di durata circa Δ ad ogni transizione di *x*
- Nessuna delle precedenti

6) Se devo dividere +4602 per -3, devo usare una IDIV con dividend:

- a 16 bit, ed il resto sarà negativo
- a 16 bit, ed il resto sarà positivo
- a 32 bit, ed il resto sarà negativo
- Nessuna delle precedenti

7) Perché nel calcolatore visto a lezione il ciclo di lettura nello spazio di I/O dura un clock in più rispetto a quello dello spazio di memoria?

- Perché si assume che i dispositivi di I/O sono più lenti dei chip di memoria
- Perché si deve dare tempo alle maschere delle interfacce di andare a regime, mentre davanti alla memoria non ci sono maschere
- Perché le letture in memoria non sono distruttive, mentre quelle nell'I/O possono esserlo indirettamente
- Nessuna delle precedenti

8) Dato $A \equiv (\beta - 1, \beta - 1, \dots, \beta - 1)_\beta$, naturale in base β ,

$$|A|_\beta = \beta - 1$$

- Vero, qualunque sia la base β
- Vero, ma solo in base $\beta = 10$
- Falso, qualunque sia la base β
- Nessuna delle precedenti

$x_1x_0 \backslash x_2x_1$	00	01	11	10
00	1	0	0	-
01	-	1	0	-
11	-	1	-	0
10	-	1	0	1

9) Il costo a diodi della sintesi a costo minimo a porte NOR della mappa sopra riportata è pari a:

- 10
- 7
- 9
- Nessuna delle precedenti

10) Sia $X = 5555$ la rappresentazione in complemento alla radice di un numero intero x in base 8. Ciò significa che x è un numero

- positivo, rappresentabile anche su tre cifre
- positivo, ma non rappresentabile su tre cifre
- negativo, rappresentabile anche su tre cifre
- negativo, ma non rappresentabile su tre cifre



#	High-level	Solution
1	I need to know how MOV moves data into registers (in what order)	<p>→ We're working in little-endian so the least significant byte is stored in the lowest address</p> <p>→ Smallest + i = smallest + i, iterated for each 9 bit memory address</p>
2	Need to understand the conditions for a reducible integer, and the arithmetic of reducible numbers	<p>→ Definition of a reducible integer in Anki</p> <p>→ Worst case scenario is the addition between natural numbers, which works</p> <p>→ Given all the <i>other bases</i> can be represented in base 2, if it works in base 2, it works in all</p>
3	Need to understand how SAR, SHR, ROL, etc. work	<p>→ Stiamo ommettendo il sorgente quindi si fa solo 1 volta</p> <p>→ ROL takes the last bit and puts it in both CF and the first bit and shifts everything to the left</p> <p>→ <i>Nothing is conserved</i> given that we're <i>not using the CF flag</i> for intermediary stuff</p>
4	Need to know how latch SR's work	<p>→ Can't change variables at the same time in microprocessors \Rightarrow we're going to pass through an intermediary state in which we don't know what will happen</p>

5	Need to know how an XNOR gate works	→ XNOR only provides 1 if both the inputs are the same \Rightarrow generatore di impulso
6	Need to understand how IDIV works	→ Relationship with dividendo and divisore bit sizes, if dividendo is 16 bits \Rightarrow the divisore was 8 bits → See if the quoziente is representable on 8 bits → IDIV also does not obide by univoco condition, it just does truncation
7	Really learn the structure of a calculator which is really important knowledge	→ TODO
8	Need to know what the notation means	→ the $A \equiv \beta - 1, \beta - 1, \dots, \beta - 1$ means that we just have $\beta - 1$ in each position → Then the $ A _\beta = \beta - 1$ means the value I think in decimal → Doing the math it comes out to be true and it'll be true in all bases
9	This is elementary mappe di karnaugh stuff	→ Sintesi a porte NOR you flip then re-flip, just look at the ingressi because you don't optimize those
10	Problem on the riducibilità of numbers, goes back to arithmetic	→ First see if the first cifra is $> \frac{\beta}{2}$ to see if it's negative → Look at the representability of individual cifre in the reduced form → Or see the condition of the reresetability by still looking at the most significant bit and see if the last and penultimate bit are equal to each other

Domande di Reti Logiche – compito del 08/01/2025

C'è una sola risposta corretta per ogni domanda
Usare lo spazio bianco sul retro del foglio per appunti, se serve

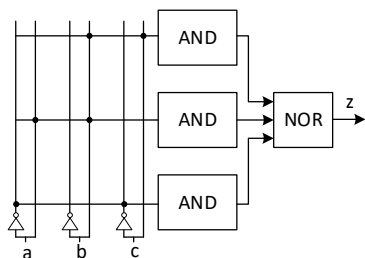
- 1) Il fatto che il risultato di una IMUL non sta sul numero di bit dove la IMUL intende scriverlo:
- È rilevato da un'eccezione
 - È indicato dal fatto che OF va ad 1
 - È indicato dal fatto che CF va ad 1
 - Nessuna delle precedenti

- 2) Un sommatore ad una cifra in base 10 BCD può essere realizzato concatenando 4 full adder in base 2 in montaggio ripple carry
- Vero, sia per naturali che per interi
 - Vero, ma solo per naturali
 - Falso
 - Nessuna delle precedenti

```
var1: .WORD 0x1020, 0x32AB
var2: .LONG var1+2
var3: .BYTE 0x66
```

- 3) Data la dichiarazione di sopra, qual è il contenuto del byte di memoria di indirizzo var2?
- 0xAB
 - 0x32
 - 0x66
 - Nessuna delle precedenti

- 4) Affinché la divisione intera tra a (dividendo) e b (divisore) abbia un quoziente rappresentabile sul numero di cifre richiesto, il fatto che lo abbia la divisione naturale tra $|a|$ e $|b|$ è condizione:
- Sufficiente
 - Necessaria
 - Necessaria e sufficiente
 - Nessuna delle precedenti



- 5) La rete disegnata di sopra riconosce un numero di stati di ingresso pari a:
- 6
 - 4
 - 3
 - Nessuna delle precedenti

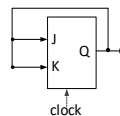
- 6) Nel calcolatore visto a lezione, un'interfaccia per la conversione D/A appare come:

- Un'interfaccia di ingresso con handshake
- Un'interfaccia di ingresso senza handshake
- Un'interfaccia di uscita senza handshake
- Nessuna delle precedenti

```
testFO:    PUSH %AL
           IN  TSR_offset, %AL
           AND $0x20, %AL
           JNZ testFO
           POP %AL
           OUT %AL, TBR_offset
           RET
```

- 7) Il sottoprogramma scritto sopra:

- Può essere chiamato per scrivere un dato in un'interfaccia di uscita con handshake
- Può essere chiamato per leggere un dato in un'interfaccia di ingresso con handshake
- Non termina mai, oppure viola l'handshake
- Nessuna delle precedenti



- 8) Il flip-flop JK montato come in figura ha un'uscita Q che:

- Cambia ad ogni clock
- Cambia al massimo una volta
- Oscilla in modo non prevedibile
- Nessuna delle precedenti

- 9) In complemento alla radice in base $\beta=14$ su una cifra, la rappresentazione dell'intero -8:

- È codificata come 1000
- È codificata come 1010
- È codificata come 0110
- Nessuna delle precedenti

- 10) Sia dato un D-latch inizializzato ad 1 al reset asincrono. La sequenza di ingressi fornita è dc=00, 10, 11, 01, 10, 11, 00. l'uscita q alla fine:

- Oscilla in modo incontrollato
- Si stabilizza su un valore casuale non prevedibile a priori
- Resta nella fascia di indeterminazione
- Nessuna delle precedenti



#	High-level	Solution
1	Gotta know which flags are impacted by assembler operations	→ In IMUL & MUL both CF e OF vengono modificati → Just fyi, in IDIV & DIV, viene generato un'interruzione interno
2	If the result needs to be in BCD, it doesn't work	→ Just using 4 full adders, i would get the correct answer in bits → However, the representation of the result in BCD would require adding +6 to the answer
3	Knowing what variables are in assembler.	→ Variables are just memory addresses → The problem has var2 the address of the third byte pointed by var1 → We don't know what the address is though
4	Understanding the relationship between numeri naturali and interi. Understanding the meaning of condizione sufficiente e necessaria	→ Divisione intera is a special case of divisione naturale from a representation standpoint → \Rightarrow condizione necessaria
5	Understand how gates work → write out the tabella di verità → see how many states are 'riconosciuto' \Rightarrow uscita == 1	→ Following the steps above I get to 3
6	Know more about the struttura del calcolatore	→ Full answer is literally written in the notes, <i>interfaccia di uscita senza handshake</i>

#	High-level	Solution
7	Need to understand how the flags work, and how the AND operation works	<p>→ We want FO to be 1 in order to be able to write something new</p> <p>→ AND replaces in the operando destinatario the <i>result</i> of and AND gate of each bit</p> <p>→ The important part is that we have JNZ which means that if the FO flag is 1, we stay in the loop, which isn't correct</p>
8	Need to understand how the flip flop JK works.	<p>→ ANKI for flip-flop JK</p> <p>→ Changes at most one time based on how the flip-flop JK works</p>
9	Need to know how complemento alla radice works	<p>→ -8 is out of the range of representable numbers in complemento alla radice in base $\beta = 14$</p>
10	Need to know how a D-latch works, and an important physical detail where variables do not change simultaneously	<p>→ The d-latch passes over 10 or 01 when going from 11-00, so we don't know the last value</p>