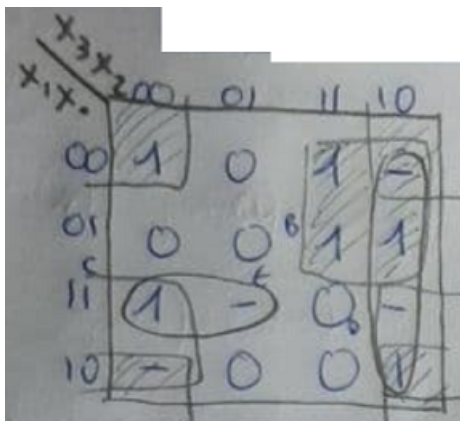


# Domande orali Reti logiche

## 1 Sessione invernale A.A 2020-2021

### 1.1 Primo appello

1. Struttura di un D flip-flop partendo da un D latch. Temporizzazioni, significato di rete trasparente, montaggio master-slave.
2. Fase di esecuzione di Jcond.
3. Interfaccia seriale: visione funzionale, struttura interna, descrizione del trasmettitore
4. Sintesi porte nand di una mappa di Karnaugh.



5. Ipotesi preliminari e condizioni di fattibilità della divisione intera in qualunque base. Disegno del circuito

### 1.2 Secondo appello

1. Sintetizzare la rete che converte un numero naturale da base 2 a base 10. Si supponga che il numero da convertire sia minore di 256.
2. Fase di fetch dell'istruzione:  

```
MOVB $operando, %AL
```
3. Conversione analogico-digitale: principi di funzionamento ed errori. Schema del convertitore A/D visto a lezione e dell'interfaccia di conversione A/D. Sottoprogramma per l'ingresso di un valore dall'interfaccia di conversione A/D.
4. Procedimento per individuare forma canonica SP, mintermine, implicante, implicante principale, IP essenziale, IP assolutamente eliminabile.
5. Memoria RAM statica: schema, temporizzazione dei cicli di lettura e di scrittura. Connessione al bus del calcolatore visto a lezione di due moduli di RAM 64kx4 bit, in modo che implementino le celle di memoria a partire dall'indirizzo 'H120000.

### 1.3 Terzo appello

1.
  - Disegnare la struttura interna del full adder
  - Sintetizzare un circuito sottrattore a due cifre in base 2 utilizzando un full-adder e una cifra, completo dell'uscita di overflow.
  - Rappresentare le uscite quando gli ingressi sono  $b_{in} = 0, X = 10, Y = 01$
2. Contatore espandibile ad una cifra in base 3. Sintetizzare fino alla ottimizzazione della parte combinatoria. Cosa conta il contatore?
3. Dato il processore visto a lezione, scrivere la fase di fetch di un nuovo formato di istruzioni (si assuma che esista un terna di bit utilizzabile per indicare un nuovo formato). Le istruzioni sono del tipo

OPCODE \$byte, indirizzo

4. Realizzare un flip-flop D-positive-edge triggered partendo da un flip-flop JK.
5. Sintetizzare l'unità XXX disegnando i circuiti della parte operativa e della parte controllo<sup>1</sup>. Riempire il diagramma temporale che ne mostra l'evoluzione<sup>2</sup>

```
module XXX(clock, reset_);
    input clock, reset_;
    reg A,B, STAR;
    parameter S0 = 0, S1 = 1;

    always @(reset_ == 0) begin A <= 0; B <= 0, STAR <= S0; end
    always @(posedge clock) if(reset_ == 1) #3
        casex(STAR)
            S0: begin A <= B+1; B <= A; STAR <= (A==1)? S1: S0; end
            S1: begin B <= A; STAR <= (B == 0) ? S1 : S0; end
        endcase
endmodule
```

---

<sup>1</sup>Specifico che è stato un po' ambiguo, voleva anche Verilog.

<sup>2</sup>Ci ha dato una foto di esempio di diagramma, lo schema era simile a quello dei pdf di Corsini