

# 1 Lezione del 07-11-24

## 1.1 Reti sequenziali sincronizzate

Le reti sequenziali sincronizzate (RSS), a differenza della asincrone (RSA), non si aggiornano per la sola variazione degli ingressi, ma per l'arrivo di un determinato segnale periodico, che chiamiamo **clock**.

Il clock è un segnale con forma d'onda periodica, di frequenza  $\frac{1}{T}$  periodo, e *duty cycle* (ciclo di lavoro)  $\frac{\tau}{T}$  intorno al 50%. Solitamente l'evento di sincronia delle reti sequenziali sincronizzate è il **fronte di salita** del clock.

## 1.2 Registri

Un registro a  $W$  bit è una collezione di  $W$  D flip-flop positive edge-triggered, che hanno:

- $W$  ingressi  $d_i$  e  $W$  uscite  $q_i$  separate (in verità ricordiamo che troviamo sempre  $q$  e  $\bar{q}$  negata, noi riporteremo solo la prima per semplicità);
- Un ingresso  $p$  comune.

Si ha che  $p$  funge da **segnale di sincronizzazione**.

### 1.2.1 Pilotaggio di registri

Per il corretto pilotaggio di un registro gli ingressi  $d_i$  devono essere stabili intorno al fronte di salita del clock, per un tempo  $T_{setup}$  prima e  $T_{hold}$  dopo il fronte stesso. L'uscita cambia dopo, come avevamo visto per i D flip-flop, un tempo  $T_{prop} > T_{hold}$ .

Tutto ciò che accade in ingresso fra due istanti di sincronizzazione è irrilevante e non viene memorizzato.

Il registro *memorizza* lo stato di ingresso al **fronte di salita**. Gli stati di ingresso fra due fronti di salita adiacenti possono essere identici, adiacenti o non adiacenti: è irrilevante in quanto, come abbiamo detto, l'aggiornamento accade soltanto nelle condizioni di stabilità intorno al fronte di salita del clock.

Dopo il fronte di salita, le uscite cambiano il loro valore dopo  $T_{prop}$ .

Possiamo quindi aggiornare la nostra definizione di RSS come *collezione di registri e reti combinatorie*, montati arbitrariamente, purchè non ci siano anelli di retroazione di reti combinatorie (costituirebbero reti sequenziali asincrone). I registri hanno tutti lo stesso clock in comune, e possono formare anelli, in quanto abbiamo visto dal loro pilotaggio, questo non genera problemi.

### 1.2.2 Regole di pilotaggio per RSS

Dato l' $i$ -esimo fronte di salita del clock al tempo  $t_i$ , lo stato di ingresso ai registri dovrà essere stabile, dalle loro regole di pilotaggio, nell'intervallo  $[t_i - T_{setup}, t_i + T_{hold}]$ . Non potrò quindi scegliere periodi  $T$  del clock piccoli a piacere: dovrò lasciare tempo ai registri di produrre nuovi valori (in tempo  $T_{prop}$ ), e alle reti combinatorie di elaborare tali valori coi loro tempi di ritardo interni, e quindi di propagarsi nuovamente fino ai registri.

Definiamo, nello specifico, i ritardi:

- $T_{in\_to\_reg}$ : il tempo di attraversamento della catena di sole reti combinatorie che da uno degli ingressi della rete all'ingresso di un registro;

- $T_{reg\_to\_reg}$ : il tempo di attraversamento fra l'uscita e l'ingresso di un registro;
- $T_{in\_to\_out}$ : il tempo di attraversamento fra un ingresso e un uscita dell'intera rete;
- $T_{reg\_to\_out}$ : il tempo di attraversamento fra l'uscita di un registro e un uscita della rete.

Dobbiamo introdurre poi i tempi  $T_{a\_monte}$  e  $T_{a\_valle}$ , cioè i tempi necessari all'utente della rete per, rispettivamente, **modificare** gli ingressi e **leggere** le uscite. Questi formano due ulteriori vincoli di pilotaggio in ingresso e in uscita.

Possiamo quindi porre le disequaglianze:

$$\begin{cases} T \geq T_{hold} + T_{a\_monte} + T_{in\_to\_reg} + T_{setup} \\ T \geq T_{prop} + T_{reg\_to\_reg} + T_{setup} \\ T \geq T_{hold} + T_{a\_monte} + T_{in\_to\_out} + T_{a\_valle} \\ T \geq T_{prop} + T_{reg\_to\_out} + T_{a\_valle} \end{cases}$$

che riguardano rispettivamente i tempi di percorrenza ingresso-registro, registro-registro, ingresso-uscita e registro-uscita.

In verità, avremo altri due ritardi di cui tenere conto:

- $T_{sfas}$ : il **massimo sfasamento** fra due clock. Visto che questo viene portato a elementi diversi, a qualche registro arriverà prima e a qualche registro arriverà dopo;
- $T_{reg}$ : se un registro è formato da  $W > 1$  bit, questi non cambieranno tutti contemporaneamente: dovremmo aggiungere  $T_{prop} + T_{reg} = T'_{prop}$ . A questo punto possiamo considerare solo  $T'_{prop}$ .

### 1.2.3 Modello di Moore e di Mealy ritardato

Potremmo voler determinare qual'è la più vincolante fra le disequaglianze riportate prima. Questa, chiaramente, è quella che copre il percorso più lungo, cioè la terza. Se decidiamo di vietare il percorso che copre, cioè quello diretto fra ingressi e uscite, otteniamo il cosiddetto **modello di Moore**: cioè, un modello di RSS dove non si ammettono reti combinatorie che collegano gli ingressi direttamente alle uscite.

Un'altro vincolo che potremmo voler rilassare è il quarto, nel cosiddetto **modello di Mealy ritardato**. Questo equivale a prelevare le uscite direttamente dalle uscite dei registri, cioè a eliminare il tempo  $T_{reg\_to\_out}$ .

## 1.3 Contatori

Un contatore è una RSS il cui stato di uscita può essere visto come un **numero naturale** ad  $n$  cifre in base  $\beta$ . Ad ogni clock, il contatore **incrementa** o **decrementa**.

Abbiamo che si può realizzare un contatore collegando un modulo sommatore a  $n$  cifre a un registro a  $n$  cifre. L'uscita del registro viene collegata in anello di retroazione a uno degli ingressi del sommatore. Impostando il  $C_{in}$  del sommatore a 1, e il suo secondo ingresso ad un'array di  $n$  generatori di costante 0, si ha un contatore **incrementatore**, cioè che incrementa il suo valore ad ogni ciclo di clock. L'equivalente **decrementatore** si può creare usando un sottrattore a  $n$  cifre invece di un sommatore.

Si può creare un contatore con ingresso di abilitazione (sostanzialmente una **variabile di controllo**), cioè che incrementa o decrementa solo se è alto un certo bit di controllo, collegando tale bit al carry (o al borrow) del sommatore (sottrattore).

### 1.3.1 Scomposizione in moduli di contatori

Un contatore può essere scomposto, in qualsiasi base, in una serie di contatori ad una cifra collegati a **catena di riporti** (*ripple carry*). In questo caso il registro è dato dalla combinazione di  $n$  registri, uno per ogni cifra (e quindi per ogni contatore), tutti sincronizzati sullo stesso clock.