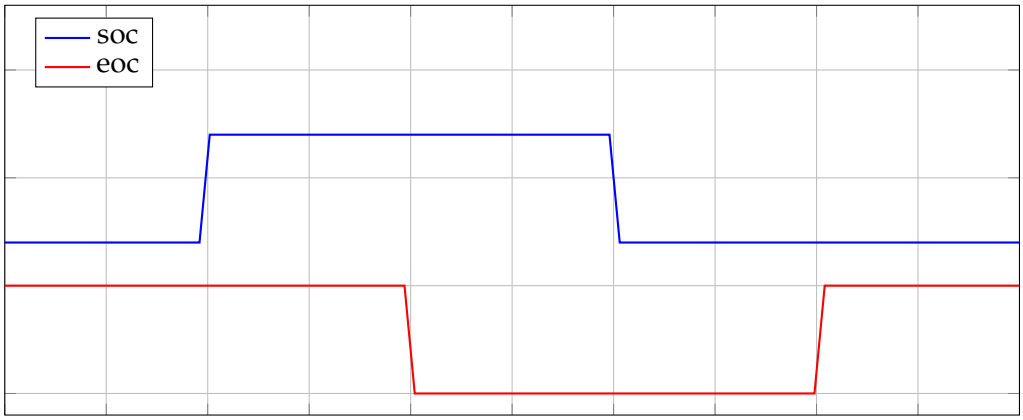
1 Lezione del 21-11-24

1.1 Handhake soc-eoc

Abbiamo visto l'handshake dav-rfd. Vediamo adesso un altro tipo di handshake, detto **handshake soc-eoc** (*Start Of Computation, End Of Computation*). Nell'handshake soc-eoc, è il consumatore a fare la prima mossa, cioè a segnalare al produttore che bisogna di un nuovo dato.

La situazione di riposo è quella dove soc è 0 e eoc è 1. In questo caso, l'ultimo dato è già sulla linea di trasmissione, quindi possiamo leggerlo liberamente. Quando il consumatore richiede un nuovo dato, mette soc a 1. A questo punto il produttore metterà eoc a 0: questo signnifica che la computazione è in corso e non possiamo leggere dalla linea di trasmissione. Una volta rilevato eoc a 0, il consumatore dovrà rimettere soc a 0, per rispettare l'alternanza dei segnali di handshake. Al rialzarsi di eoc, quindi, il nuovo dato sarà pronto e potremo leggerlo dalla linea di trasmissione, soc sarà a 0 da prima e ci troveremo quindi nuovamente nella situazione di riposo.

Viste su un grafico, le variabili soc e eoc assumono, durante un ciclo di handshake valido, i seguenti valori:



1.2 Sintesi di RSS complesse

Finora abbiamo visto descrizioni scritte attraverso il **linguaggio di trasferimento fra registri** del Verilog. Adesso dobbiamo vedere di come *sintetizzare* effettivamente una rete complessa, attraverso **circuiti elementari** noti. Notiamo che non parleremo di *sintesi ottime*, in quanto questo non è un problema che adesso ci poniamo.

Vediamo quindi come operiamo nella pratica: dovremo operare scomposizione in **parte operativa** e **parte di controllo**:

- **Parte operativa:** contiene la logica necessaria all'interfacciamento col mondo esterno e alla produzione di stati di ingresso per i **registri operativi**;
- **Parte di controllo:** si occupa di mantenere aggiornato lo stato interno.

Queste due reti hanno lo stesso clock, e comunicano fra di loro attraverso due gruppi di variabili: quelle di **comando** (PC $\rightarrow$ PO), e quelle di **condizionamento** (PO $\rightarrow$ PC).

Per realizzare la **parte operativa** dobbiamo pensare ai registri operativi come **registri multifunzionali**. Per ogni registro isoliamo le μ-operazioni diverse, una per ogni

stato, che dovremo effettuare in modo da ricavare il suo ingresso, e le inseriamo in una rete combinatoria con in coda multiplexer guidati da variabili di comando (che sono sia variabili di controllo dei multiplexer e variabili di comando della parte operativa).

Dalla parte operativa genereremo le variabili di condizionamento, attraverso un suo sottoinsieme detto **rete combinatoria di condizionamento**, che useremo per sintetizzare la parte di controllo. La rete combinatoria di condizionamento prende come ingressi le variabili di ingresso della RSS e lo stato dei registri operativi (che in generale sono visibili alla parte operativa). Notiamo che la parte operativi rappresenta effettivamente una rete di Mealy, in quanto alcune uscite escono da registi (regisri multifunzionali), e altre vengono direttamente da reti combinatorie connesse agli ingressi (variabili di condizionamento).

La **parte di controllo** si occupa invece di gestire, sostanzialmente, il registro STAR. Prende in ingresso le variabili di condizionamento appena generate, e restituisce in uscita le variabili di comando dei multiplexer della parte operativa. Questo si può implementare effettivamente come un RSS di Moore, dove le uscite dei registri rappresentano le variabili di comando stesse.

Notiamo quindi di aver ricondotto l'RSS a due reti sequenziali, una di Moore e una di Mealy, in retroazione fra di loro. Questo, come abbiamo visto, si può fare in quanto una di loro è di Moore.