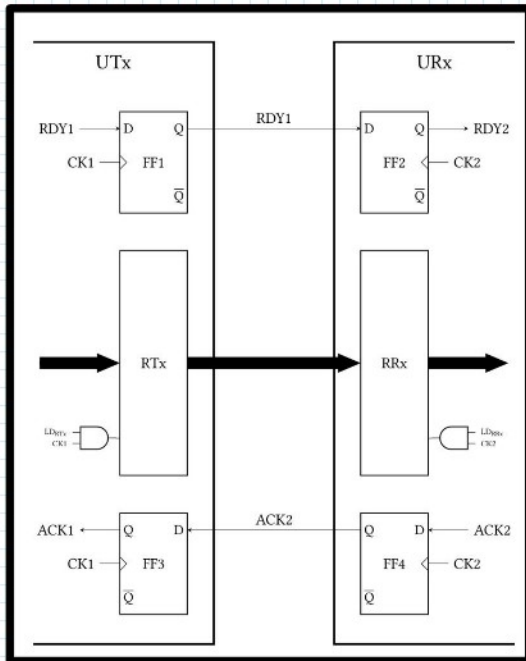


# PROTOCOLLO DI HANDSHAKING

È UN INSIEME DI SEGNALI E DI REGISTRI TAMPONE che permettono a due reti combinatorie che viaggiano a velocità differenti, di parlarsi: ossia i dati rimangono stabili per tutto il periodo di clock!

Abbiamo un dispositivo trasmittente e un ricevente, che vogliono scambiarsi dei dati unidirezionalmente;



TRASMITTENTE E RICEVENTE VANNO A VELOCITÀ DIVERSE;

IL TRASMITTENTE DEVE ESSERE INFORMATO CHE IL RICEVENTE ABBIÀ RICEVUTO QUELLO DATO.

IL RICEVENTE DEVE SAPERE CHE I DATI SONO STATI PRODOTTI DAL TRASMITTENTE.

Abbiamo bisogno di segnali di controllo dedicati che collegano questi dispositivi PER FARE SÌ CHE IL RICEVENTE RICEVA UN DATO che è STATO PRODOTTO DAL TRASMITTENTE, IL RICEVENTE PRELEVA QUEL DATO E IL TRASMITTENTE NON FA NULLA FINCHÉ NON È STATO INFORMATO DELLA LETTURA COMPLETA, QUANDO LA LETTURA È STATA COMPLETATA IL RICEVENTE DICE AL TRASMITTENTE CHE HA LETTO I DATI.

Quindi ABBIAMO BISOGNO ALMENO DI: due segnali di controllo!

- 1) SERVE A COMUNICARE AL TRASMITTENTE CHE IL DATO È STATO LETTO!
- 2) SERVE A COMUNICARE AL RICEVENTE CHE UN NUOVO DATO È PRONTO!

CHE SONO I SEGNALI DI READY E ACKNOWLEDGMENT: **RDY, ACK!**

IL segnale di **Ready** viene utilizzato per dire al Ricevente che c'è un nuovo dato!  
 IL segnale di **ACK** serve a dire al trasmittente che quel dato è stato letto!

Siccome sia trasmittente che ricevente viaggiano su clock differenti, dobbiamo mantenere stabile quei segnali fin tanto che non finiscono quello che devono fare.

Appena il mio produttore comunica che un altro dato è pronto, scriverà 1 sul FLIP-FLAP, e questa scrittura avviene alla velocità del clock-1, cioè il suo!

E scriverà 1.

A questo punto, quell'uno rimane e diventa stabile

Ora seguendo il clock del ricevente, si legge il valore che era stato precedentemente scritto sul FLIP-FLAP e se lo salva IN RDY2.

DAL quel momento in poi rimane stabile!

Se il trasmittente prima di scrivere nel flip flop aveva scritto i dati da voleva trasmettere all'interno di RTx (registro tampone), quando il ricevente legge Ready2 può ABILITARE



SE IL TRASMETTENTE PRIMA DI SCRIVERE NEL FLIP FLOP AVEVA SCRITTO I DATI CHE VOLEVA TRASMETTERE ALL'INTERNO DI  $RT_x$  (REGISTRO TAMPONE), QUANDO IL RICEVENTE LEGGE Ready2 PUÒ ABILITARE IL WRITE ENABLE SU QUELLO REGISTRO  $RR_x$ .

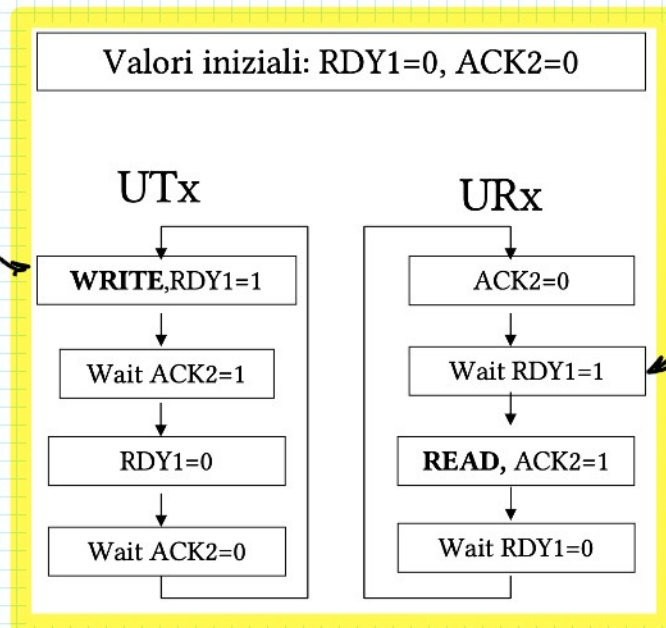
**Cosa mi garantisce che i dati non vengano SOVRASCRITTI mentre stanno passando dal Registro sorgente a quello destinazione? IL fatto che stiamo mettendo IN PIEDI un protocollo di HANDSHAKING!**

QUANDO IL RICEVENTE HA FATTO LA COPIA DEI DATI SCRIVERÀ 1 NEL FLIP FLOP IN  $ACK2$ , E LO MANDA AL TRASMETTENTE CON  $ACK2$ , COSÌ SA CHE IL DATO È STATO CONSUMATO!

SE STIAMO PARLANDO DI SEGNALI DI CONTROLLO ALLORA C'È UNA **MACCHINA A STATI**. C'È BISOGNO DI UN'UNITÀ INTERNA CHE GESTISCA QUESTO PROTOCOLLO, SIA NEL RICEVENTE CHE NEL TRASMETTENTE!

IL TRASMETTENTE SCRIVE I DATI ALL'INTERNO DEL REGISTRO E SEGNALE  $ready1$  UGUALE A 1 CHE HO SCRITTO I DATI;

**INIZIALMENTE L'ACK2 = 0.**



LI ASPETTA CHE IL SEGNALE DI  $READY$  VALGA 1, E VARrà 1 UN PO' DI TEMPO DOPO CHE  $UT_x$  LO HA SCRITTO!

QUANDO  $ready$  VALE 1 SIGNIFICA CHE LA SCRITTURA È STATA COMPLETATA! E COSÌ IL RICEVENTE PUÒ LEGGERE. SEGNALE DI AVERE FINITO LA LETTURA CON  $ACK2=1$ .  
E RESETTA I SEGNALI!

**VEDIAMO CHE SUCCEDEREBBE SE VADO A RIMUOVERE UNA DI QUESTE 4 ATTESE:**

