

RETI LOGICHE

PROVA SCRITTA DELL'APPELLO DEL 18-7-2005

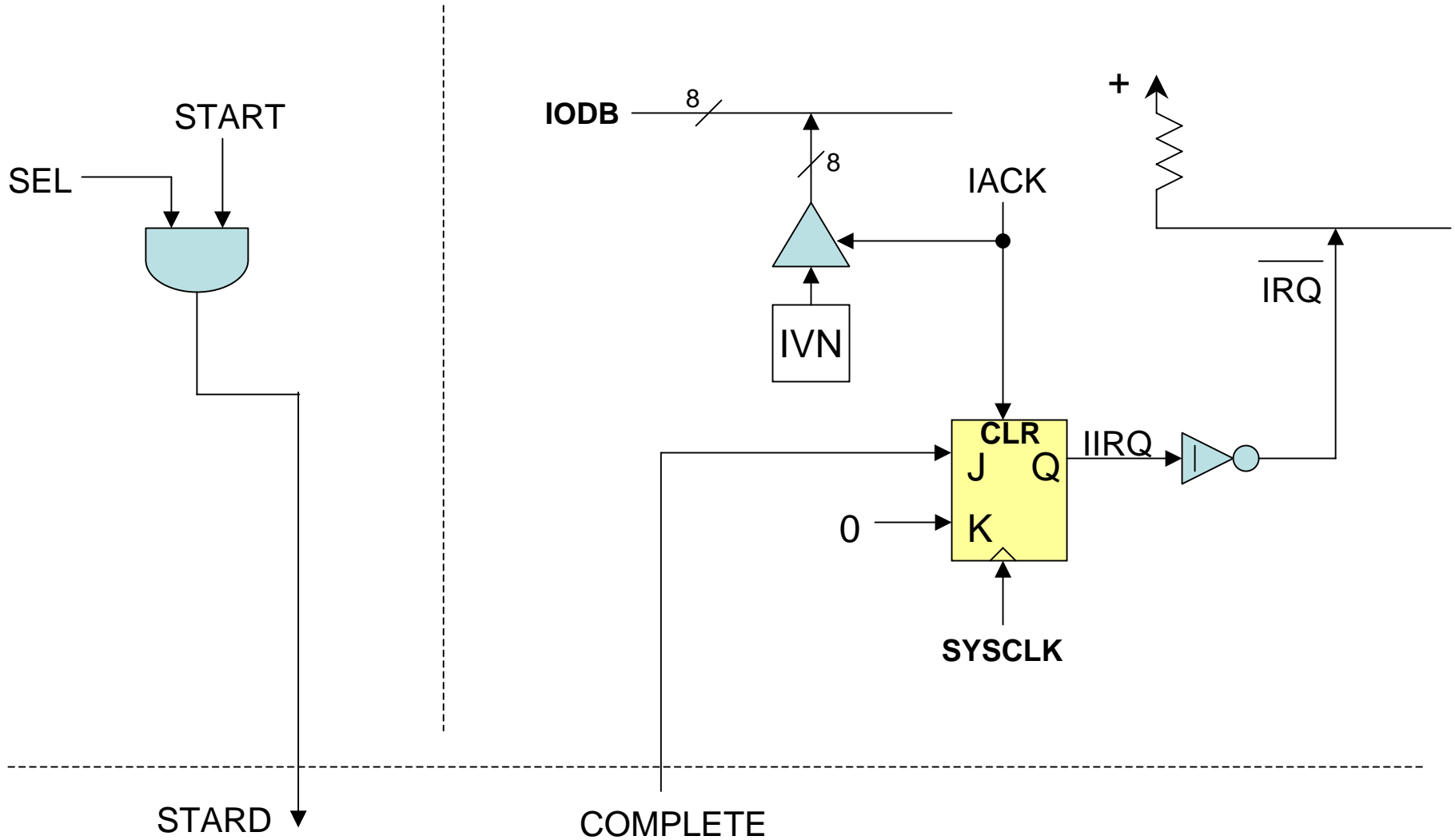
STUDENTE: _____ DOCENTE: _____

D1 (50%) - Si vuole progettare una periferica di un processore per crittografare e trasmettere su una linea seriale esterna i dati di un messaggio caricato dal micro in una memoria FIFO di interfaccia da 1 Kbyte. La periferica su comando del micro scarica i dati dalla FIFO, serializzando ciascun byte non a partire dal primo bit, ma dal bit in posizione h e poi esplorando il byte in modo circolare; a questo scopo nella periferica è incorporata anche una ROM da 1 Kbyte i cui tre bit meno significativi nella cella j -esima codificano la posizione h del bit trasmesso per primo all'interno del byte j -esimo nella FIFO.

Terminata la trasmissione di un messaggio, la periferica emette un segnale di interruzione al processore a trasmette una stringa di 0 sulla linea seriale.

Nel progetto della periferica è richiesta la descrizione delle relative temporizzazioni.

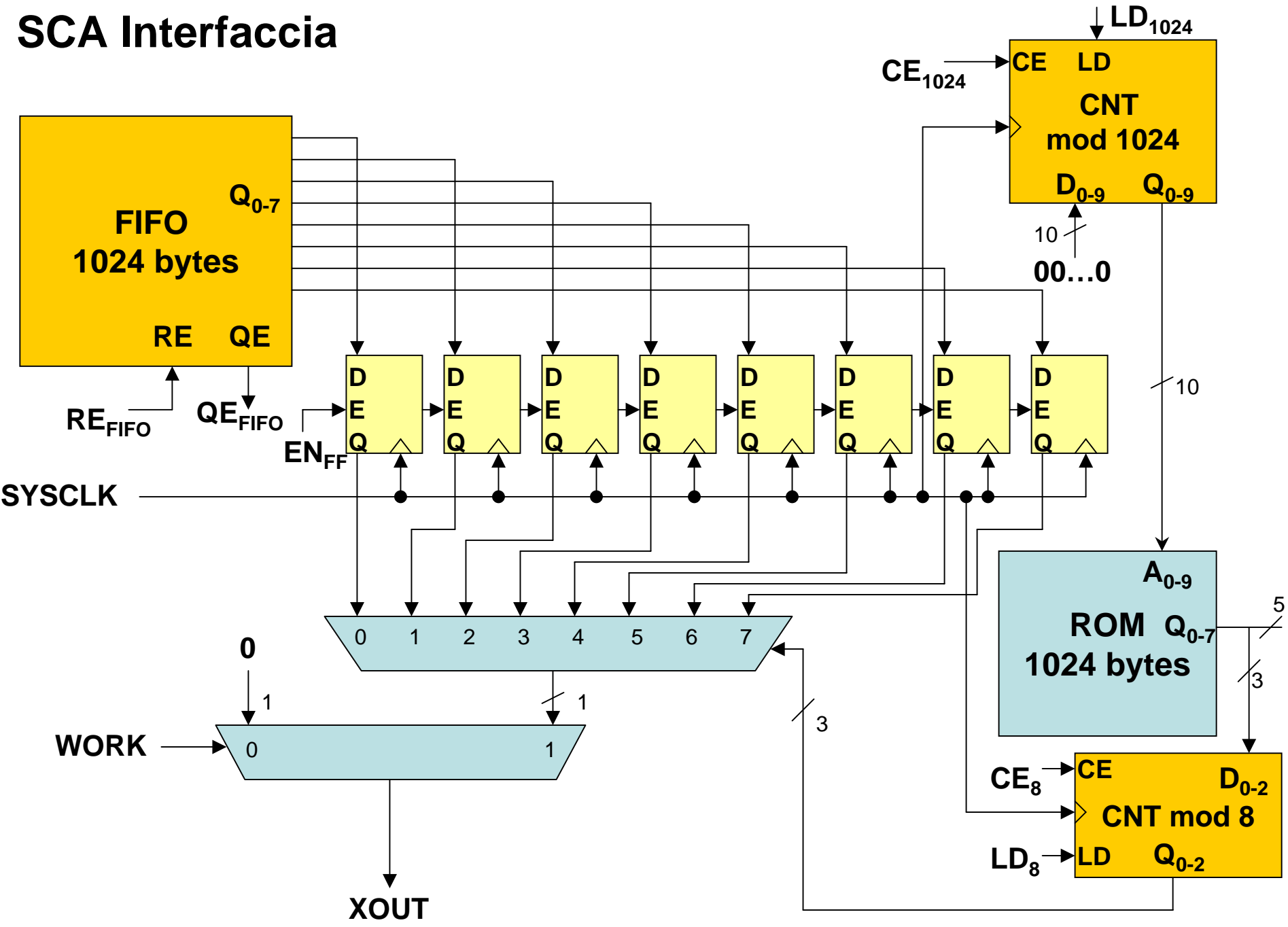
Interfaccia OUTPUT PD32 e meccanismo di INTERRUZIONE al processore



SCELTE IMPLEMENTATIVE

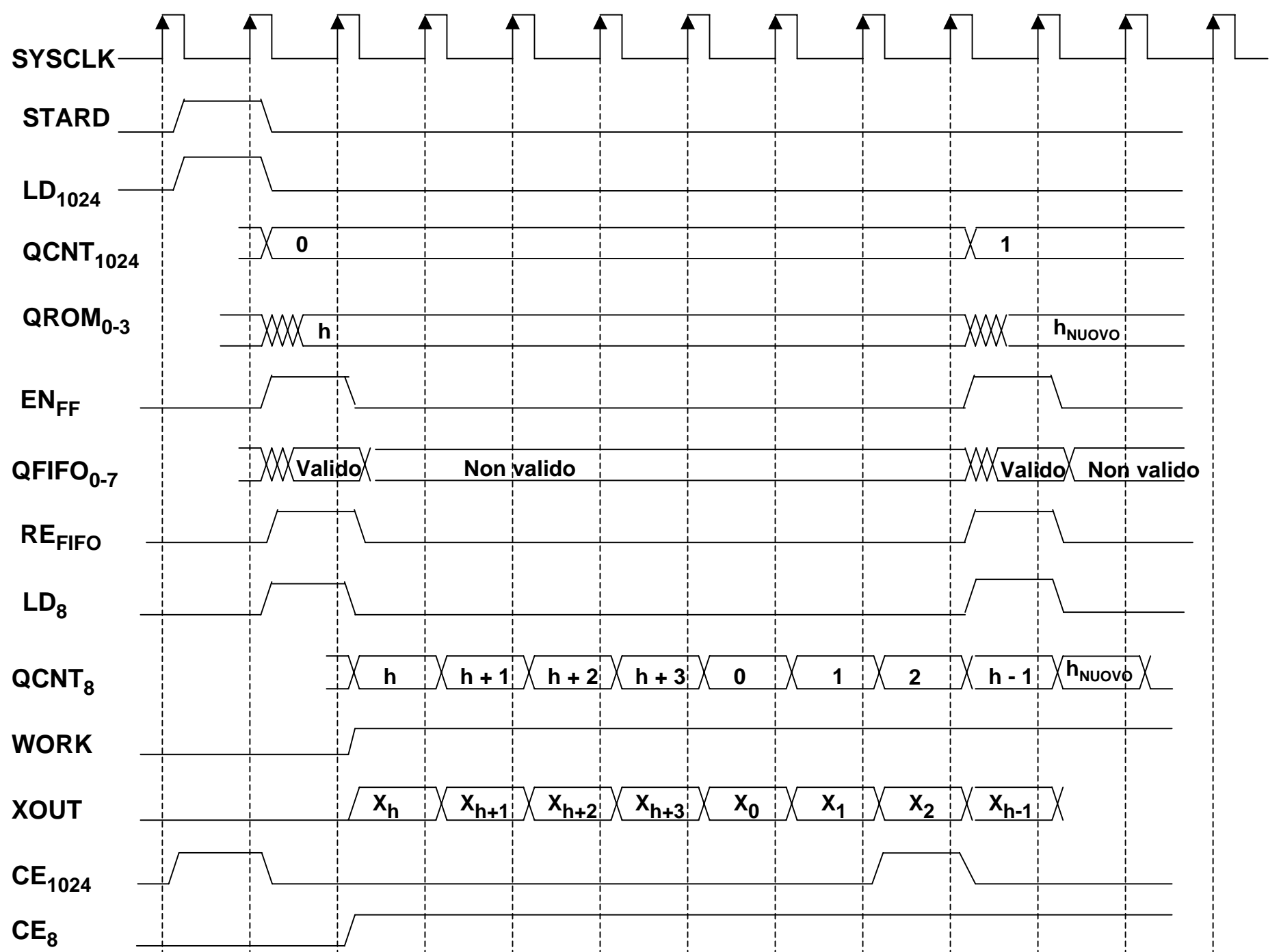
- Numero di bit di indirizzo ROM = $\log_2 1024 = 10$
- Utilizzo un **contatore mod 1024** per fornire in uscita l'indirizzo della ROM e ne viene abilitato il conteggio dopo che il dato da 8 bit è stato inviato su XOUT (quindi ogni 8 conteggi).
- Utilizzo un **contatore mod 8** settandolo al valore **h** prelevato dalla ROM (con t_{ROM} sufficientemente inferiore al periodo di system clock) e da tale valore conta circolarmente in avanti abilitando ad ogni colpo di clock un dato in ingresso ad un 8-MUX.
- Contemporaneamente al settaggio del contatore mod 8 (lettura dalla ROM) **viene letto il dato dalla FIFO** ($t_{ACCESSO}$ minore di un periodo di clock) e copiato negli 8 FF abilitando un segnale di enable EN_{FF} .
- Il segnale di **WORK** discrimina se mandare in uscita una stringa di 0 oppure i dati che man mano vengono letti dalla FIFO.
- Il segnale di STARD, opportunamente manipolato, serve per iniziare il procedimento, ed il segnale di **QE** (Queue Empty) della FIFO per terminare le operazioni.

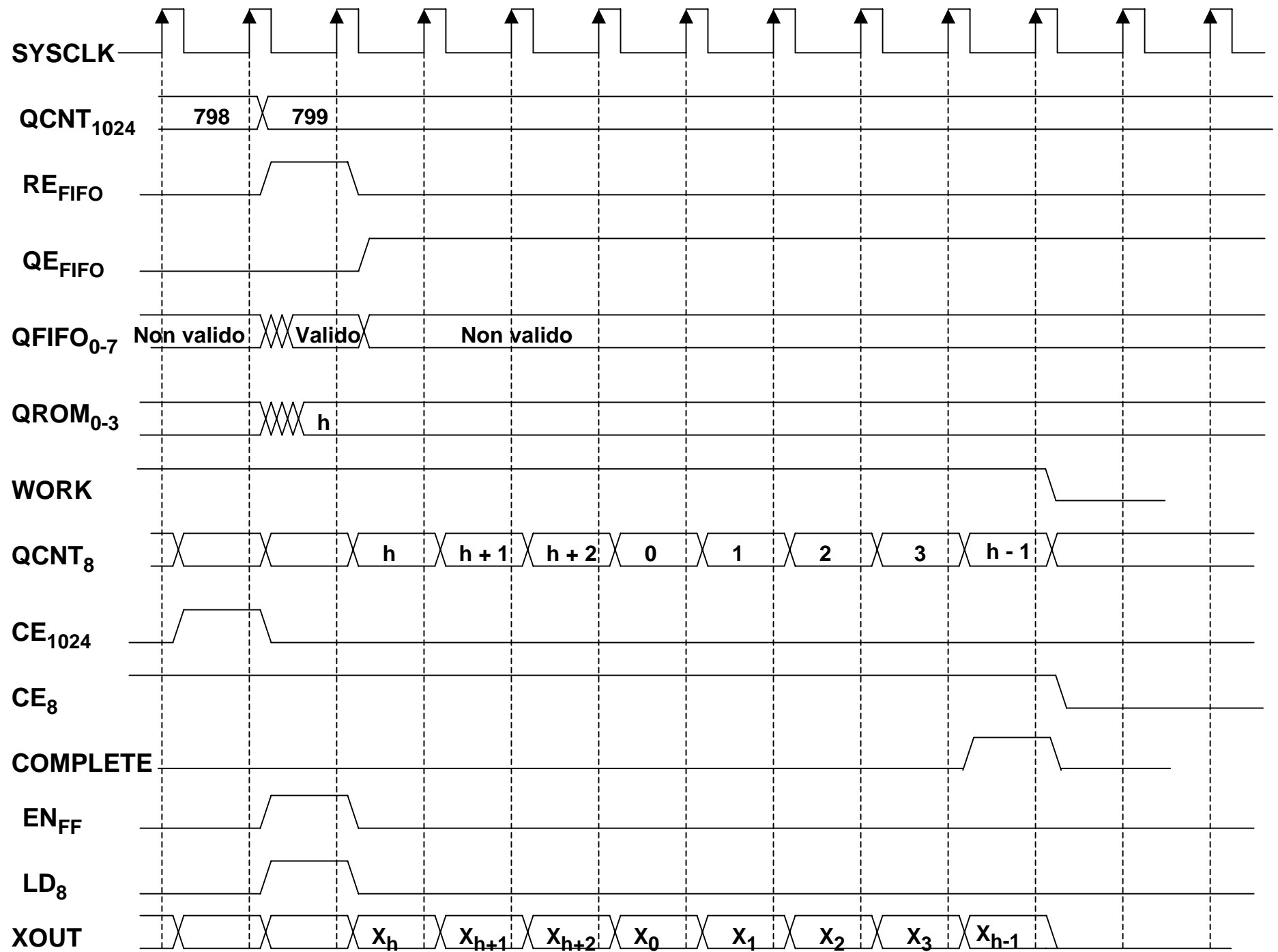
SCA Interfaccia



TEMPORIZZAZIONI

- *La prima pagina delle temporizzazioni evidenzia il meccanismo di inizio operazioni ed il normale procedimento della periferica.*
- *Il valore h per un indirizzo in ROM pari a tutti 0 viene ipotizzato posto a 4 (come esempio nelle temporizzazioni).*
- *La seconda pagina fornisce la terminazione delle operazioni (si suppone che al byte numero **799** il segnale di QE si attivi (quindi coda vuota)).*
- *Il valore h in questo caso è ipotizzato pari a 5*
- *Il **CE_g** potrebbe terminare in corrispondenza della commutazione ad **h-1** però per convenienza facciamo che termini esattamente come **WORK**.*





Considerazioni temporizzazioni / SCO

1) RE_{FIFO} , EN_{FF} , LD_8 e CE_{1024} avvengono ogni 8 conteggi, perciò sembra opportuno dotare lo SCO di un contatore modulo 8 il cui CE viene inibito quando trova QE_{FIFO} alto (è sempre inibito se Q_{STARD} inattivo: vedere tabella di verità in figura).

2) Per il CE_{1024} , siccome avviene sempre un colpo di clock prima dei RE_{FIFO} , EN_{FF} e LD_8 , possiamo mandare il TC senza ritardi (sempre che Q_{START} sia 1 e QE_{FIFO} sia 0) come CE_{1024} e gli altri tre ritardati con un FF D (supponendo che tra il LD sincrono e il CE di un contatore vinca il LD).

Q_{STARD}	QE_{FIFO}	TC	CE
0	0	0	0
0	0	1	0
0	1	0	0
0	1	1	0
1	0	0	1
1	0	1	1
1	1	0	1
1	1	1	0

3) $LD_{1024} = STARD$

4) Il TC del precedente contatore in AND con QE_{FIFO} mi da il **COMPLETE** che resetta in modo sincrono il FF_{STARD} e il FF_{WORK} .

5) I segnale di **WORK** e CE_8 vengono settati tramite il Q_{STARD} poiché dalle temporizzazioni è evidente che tali segnali divengono attivi due colpi di clock dopo l'arrivo del segnale di STARD.

6) I **RES** nei FF sono di tipo **SINCRONO**

