

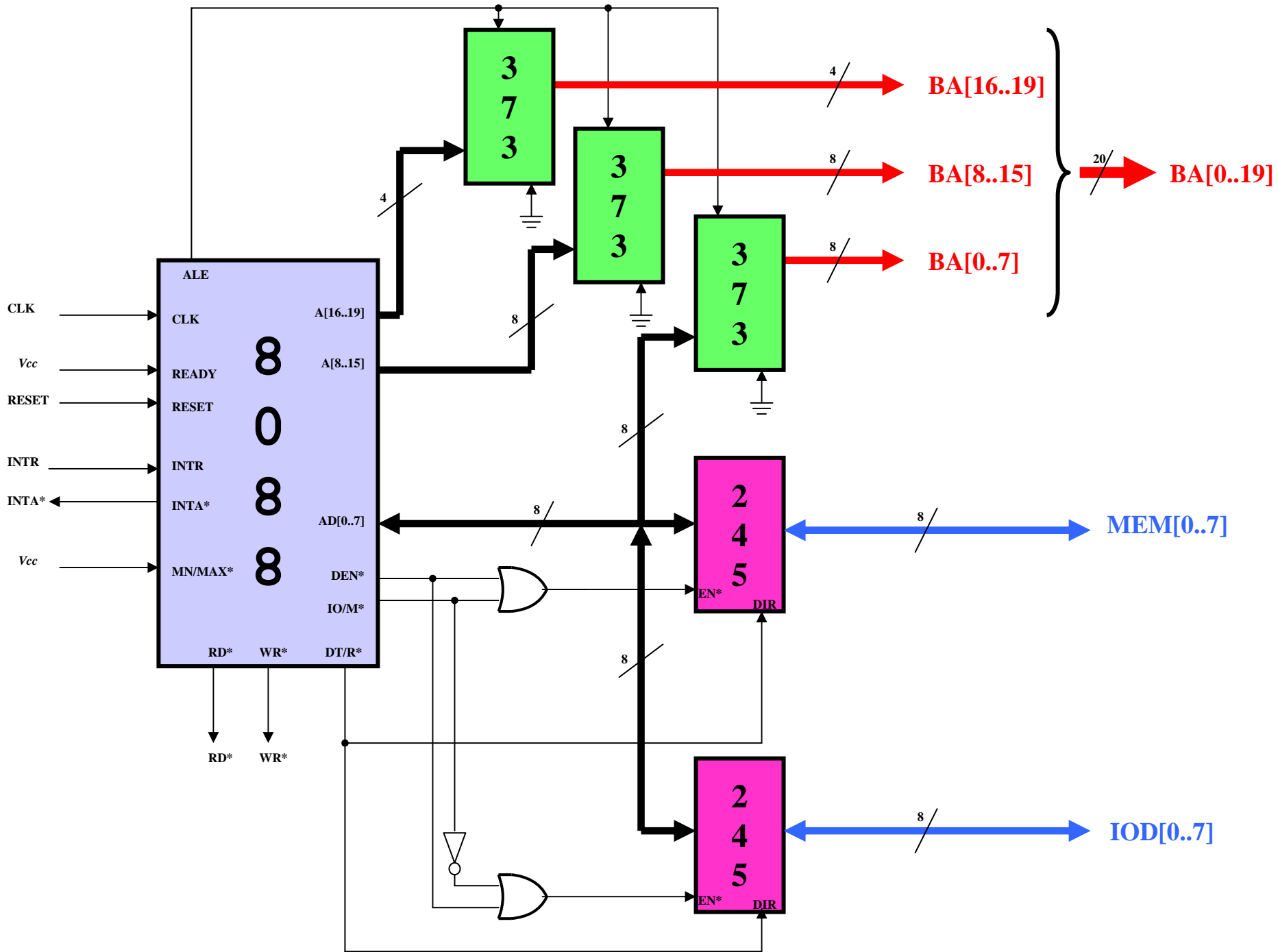
# Esercizio III-8088

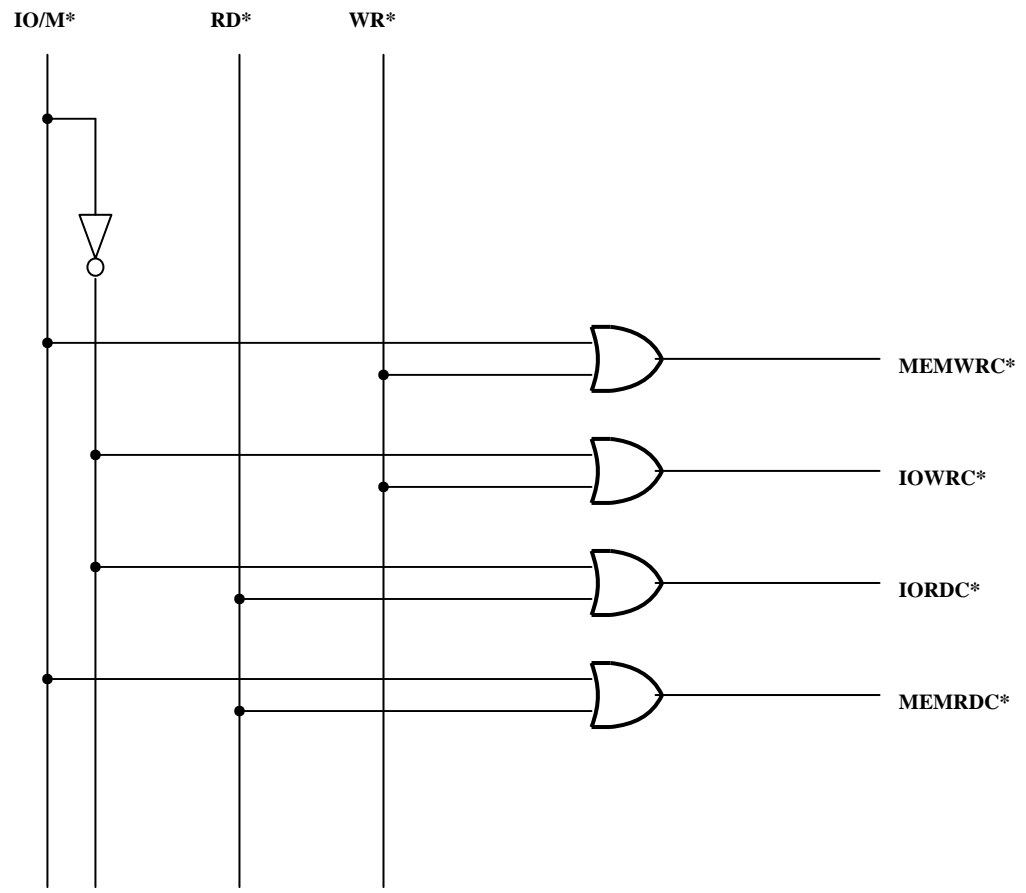
Calcolatori Elettronici LA

(13 Luglio 2005)

In un sistema con **8088** si vuole inserire i puntatori alle subroutines di risposta alle interruzioni nell'intervallo **3K-4K** anziché nell'intervallo **0K-1K**.

Progettare una rete che permetta questa funzionalità (ovvero che dopo un doppio INTA faccia leggere al processore il puntatore di memoria predisposto in tale intervallo corrispondente a quello che normalmente si troverebbe nell'intervallo all'inizio della memoria).





## Trasferimento del controllo vettorizzato nell'8088

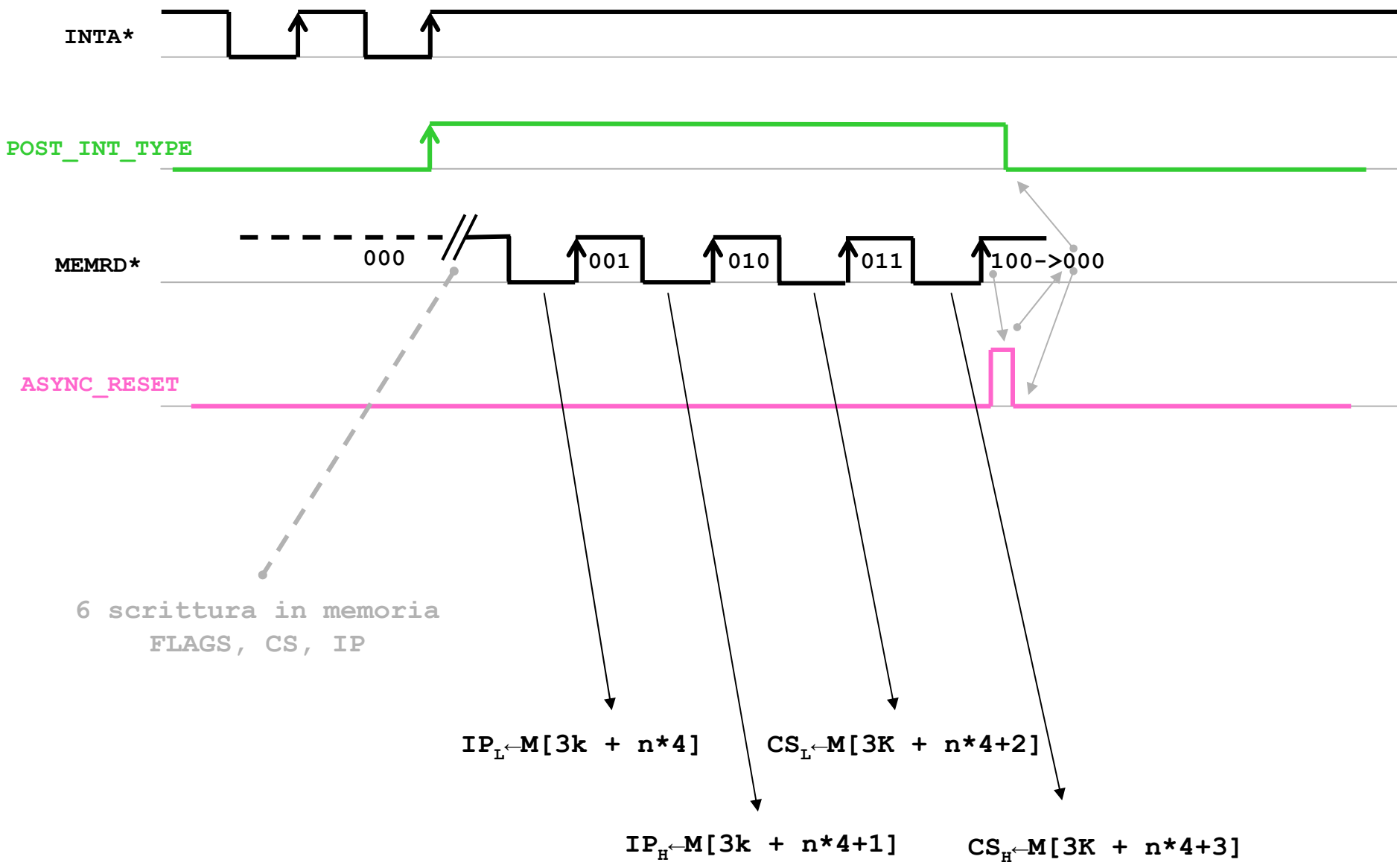
Sequenza di operazioni eseguite dalla CPU quando deve trasferire il controllo alla procedura di servizio di un interrupt:

1. **Salvataggio sullo stack del registro dei FLAGS (PUSH FLAGS)**
2. **Azzeramento del bit IF del registro dei FLAGS (CLI)**
3. **Salvataggio sullo stack di CS (PUSH CS)**
4. **Salvataggio sullo stack di IP (PUSH IP)**
5. **Lettura dell'offset della procedura di servizio da IVT e trasferimento del valore letto in IP (IP M[n\*4])**
6. **Lettura del selettore della procedura di servizio da IVT e trasferimento del valore letto in CS (CS M[n\*4 +2])**
7. *Fetch della prossima istruzione all'indirizzo di destinazione (cioè fetch della prima istruzione della procedura di servizio dell'interrupt).*

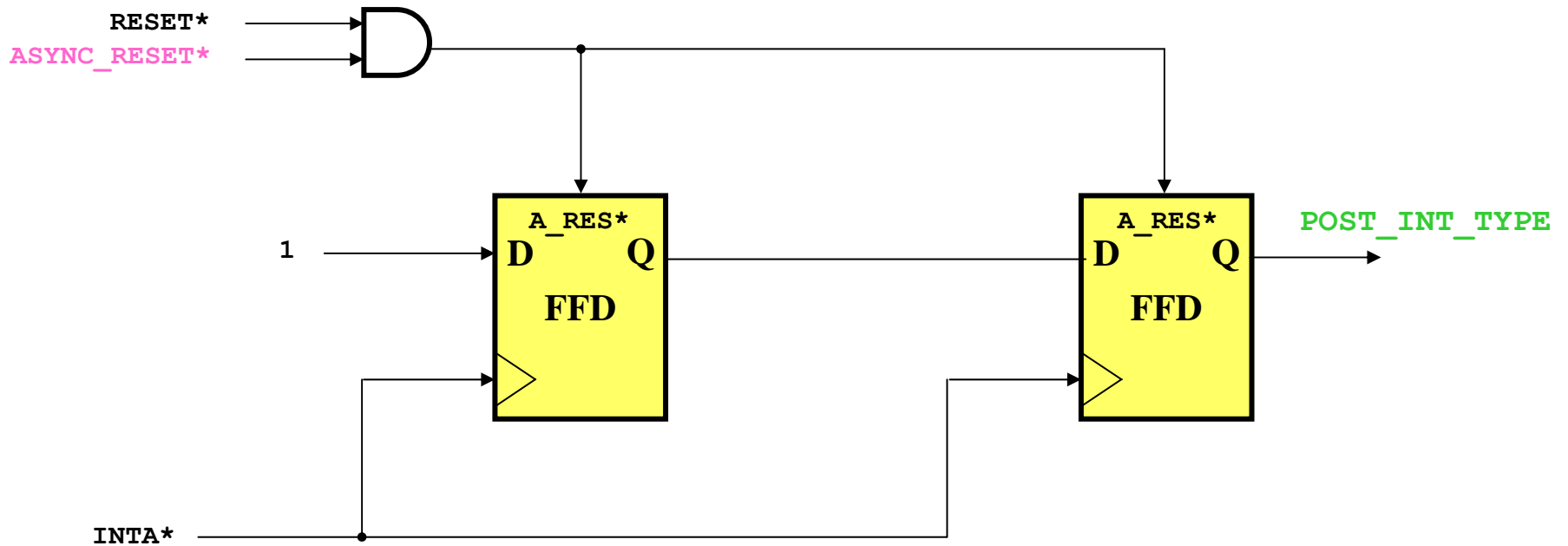
Poiché è richiesto di rimappare i puntatori alle subroutines di risposta agli interrupt dagli indirizzi **0K-1K** agli indirizzi **3K-4K** è necessario che i bit di indirizzo **A11** e **A10**, durante le 4 letture in memoria (**CS** e **IP**) eseguite dall'8088 dopo il doppio ciclo di INTA, siano settati al valore '11'.

	<b>A<sub>11</sub>A<sub>10</sub></b>		<b>A<sub>11</sub>A<sub>10</sub></b>		<b>A<sub>0</sub></b>
<b>0K-1K-1</b>	<b>00</b>	0000000000	->	<b>00</b>	1111111111
<b>3k-4k-1</b>	<b>11</b>	0000000000	->	<b>11</b>	1111111111

Durante le 4 letture in memoria necessarie per leggere l'indirizzo della routine di risposta all'interrupt (codificate dal segnale **POST\_INT\_TYPE** descritto nelle pagine successive) sarà quindi necessario sostituire agli indirizzi **A11** e **A10** (00) emessi dall'8088 il valore **11**. Poiché l'8088, **dopo avere letto l'interrupt type e prima di leggere l'indirizzo della routine di risposta all'interrupt**, salva il contenuto dei registri **FLAGS**, **CS** e **IP** sullo stack (in totale quindi 6 accessi in scrittura alla memoria) sarà necessario disabilitare il mapping degli indirizzi **A11** e **A10** durante questa fase condizionando **POST\_INT\_TYPE** con il segnale **MEMRD**.

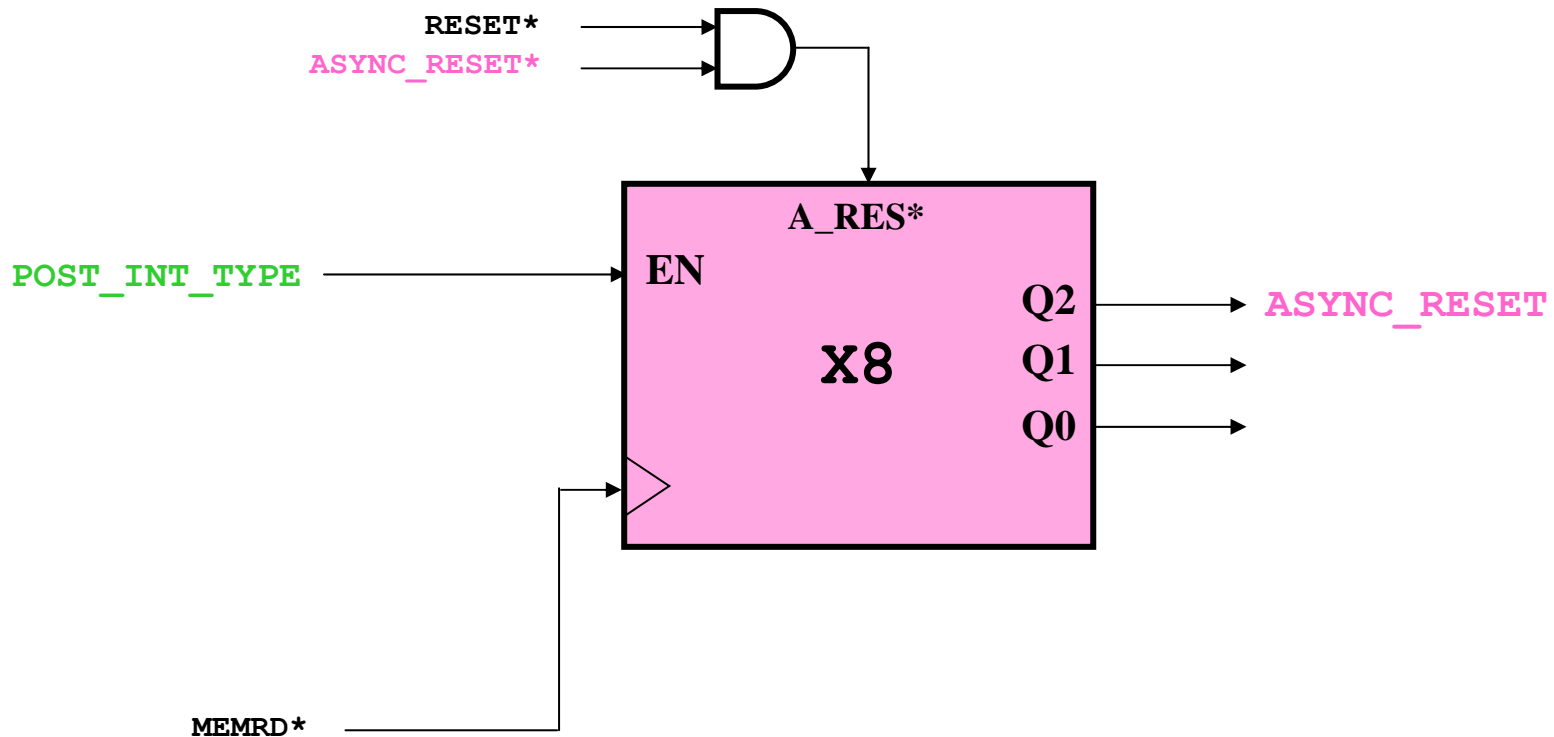


## Sintesi del segnale `POST_INT_TYPE`



Il segnale `POST_INT_TYPE` si attiva alla fine del secondo ciclo di `INTA` e viene resettato (oltre che dal segnale di `RESET` emesso all'avvio) dal segnale `ASYNC_RESET` che codifica la fine della lettura dell'indirizzo della subroutine di risposta all'interrupt. La rete che genera `ASYNC_RESET` è descritta dettagliatamente nelle pagine seguenti.

## Sintesi del segnale **ASYNC\_RESET**



Si utilizza un contatore modulo 8, dotato di comando di **ENABLE sincrono** e di **RESET asincrono** (**A\_RES**) per generare il segnale **ASYNC\_RESET** che resetta sia la rete che genera il segnale **POST\_INT\_TYPE** sia il contatore stesso dopo avere effettuato 4 letture in memoria successive al doppio ciclo di **INTA\***.

$$\mathbf{ASYNC\_RESET} = \mathbf{Q2}$$

Nella soluzione mostrata NON è possibile sostituire il contatore modulo 8 con un contatore modulo 4.

## Sintesi di BA11 e BA10

La rete seguente genera gli indirizzi BA11 e BA10 a partire dagli indirizzi A11 e A10 emessi dall'8088. Durante la lettura dei quattro byte che identificano CS e IP della routine della risposta all'interrupt (`POST_INT_TYPE·MEMRD` asserito) gli indirizzi A11 e A10 emessi dall'8088 sono sostituiti con il valore '11'.

