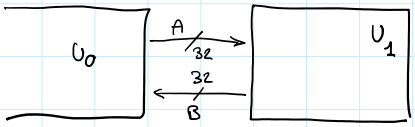
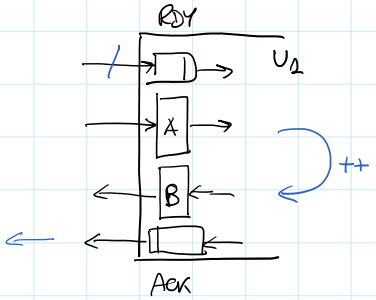
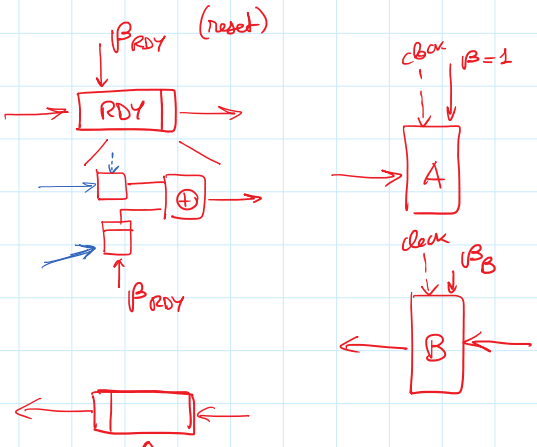


U_0 invia una richiesta e attende una risposta

U_1 in attesa "perenne" di richieste da parte di U_0

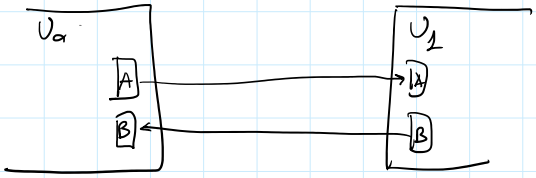
} PROTOCOLLO a "domanda/risposta"

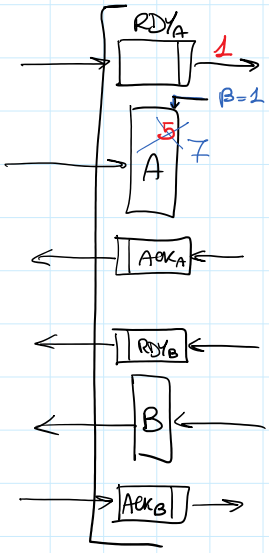


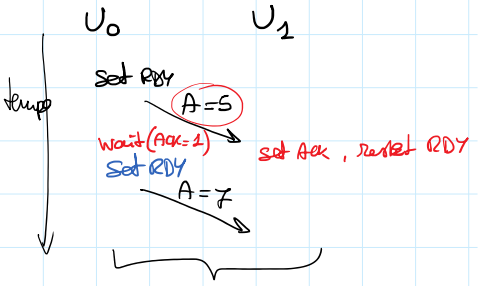


U_1 ("servente")

0. (RDY = 0) nop, 0 // attende input
(= 1) A+1 → B, // op
set ACK, // comunicazione di fine operazione verso U_0
reset RDY, // ripristino delle condizioni iniziali
0 // ricomincio il ciclo di attesa



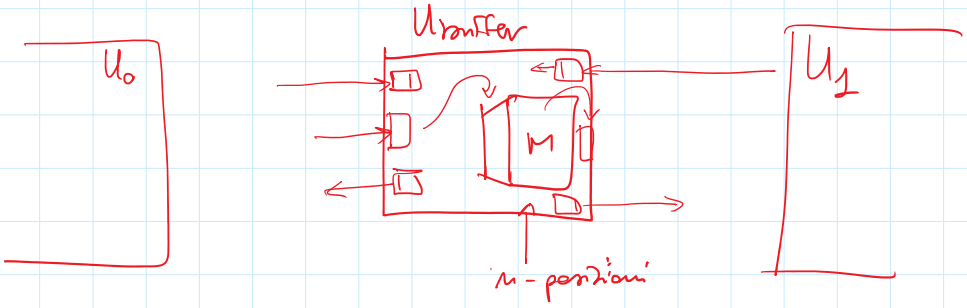




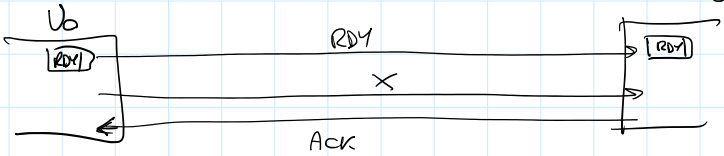
Asincrona ad 1 posizione

Asincrona a n-posizioni (buffer)

possibilità di spedire n messaggi indipendentemente dalla attività di ricezione

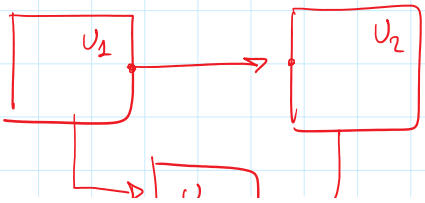


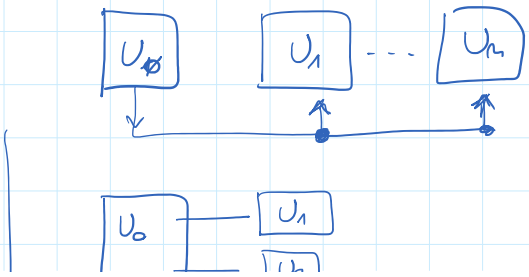
Comunicazione sincrona U_1

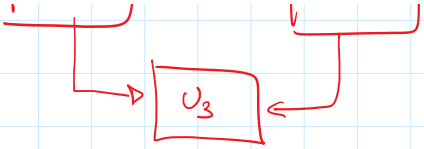


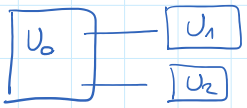
$\emptyset \rightarrow X_{out}$, set RDY, 1

1. (ACK=0) nop, 1
   (=1) reset ~~ACK~~, ...









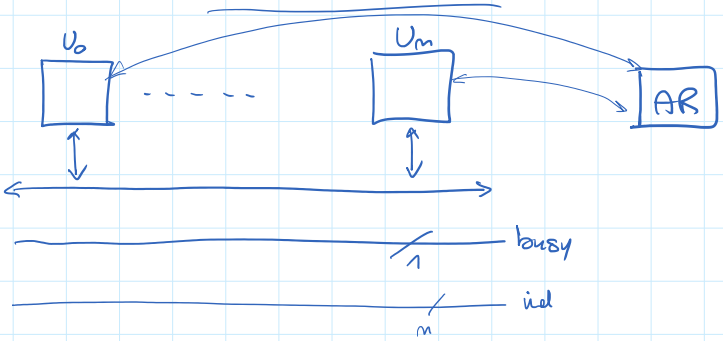
0. $X \rightarrow OUT_1$, $X \rightarrow OUT_2$
set RDY_{out_1} , set RDY_{out_2} ,

replica M volte interfacce e operazioni

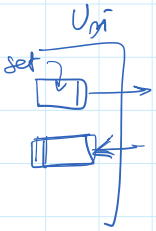
Comunicazione asimmetrica

⇓

si usano **bus** di comunicazione



1. $(ACK_1, ACK_2 = 11)$ — valore
$(= - -)$ nop, 1

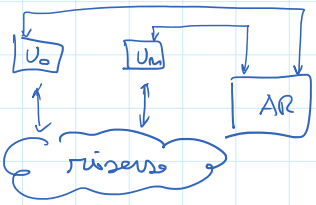


arbitro centralizzato a richieste indipendenti

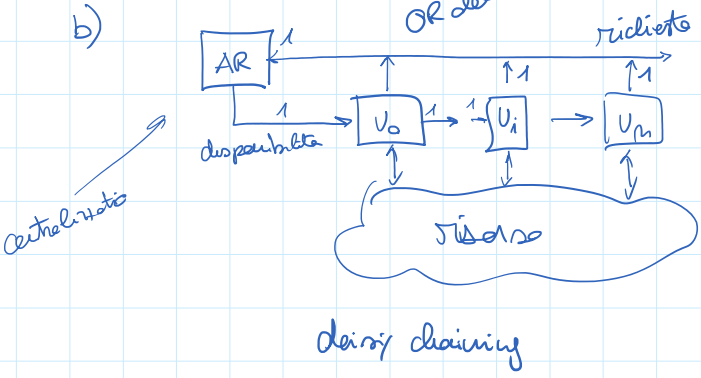
Arbitraggio

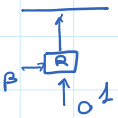
mercoledì 11 ottobre 2017 09:08

a) centralizzato a richiesta indip



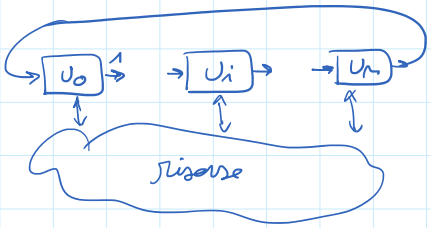
b)



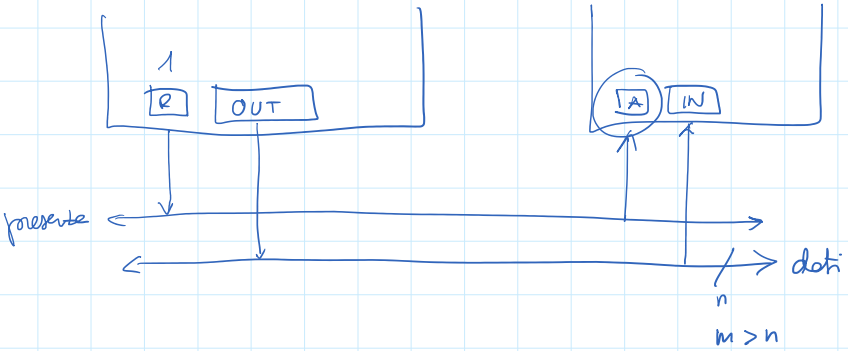


daisy chaining

c) token ring



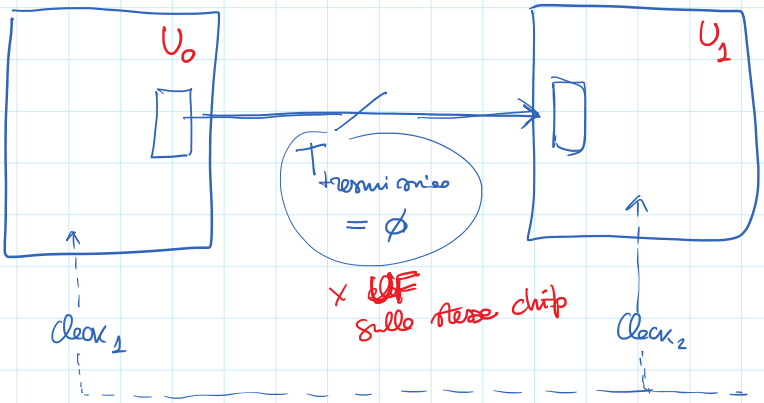
decentralizzato



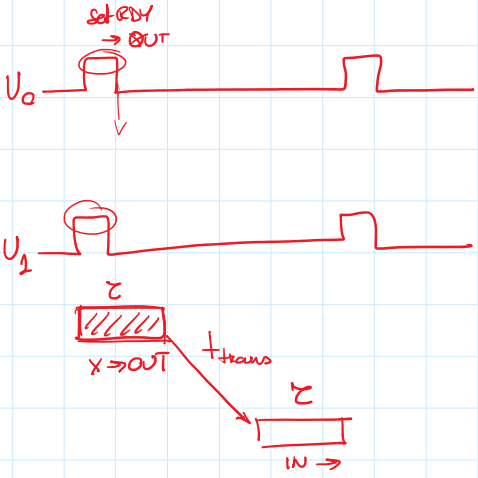
PCIe

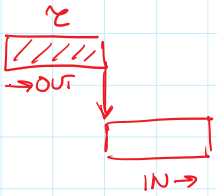
ISA

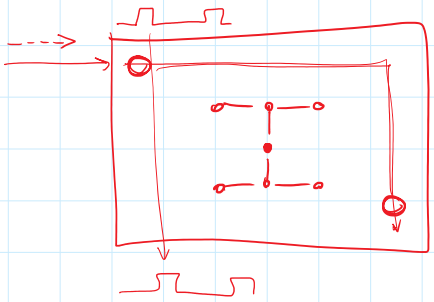
protocolli con indicatori
- "a livelli" — 4 τ (2 sul ricevente, 2 sul mittente)
- "a transizione di livello" — 2 τ (1 sul ricevente, 1 sul mittente)



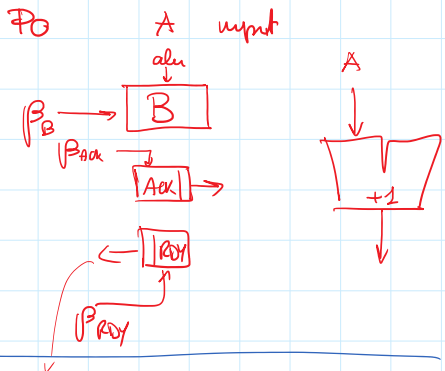
× ~~GALS~~ sullo stesso chip

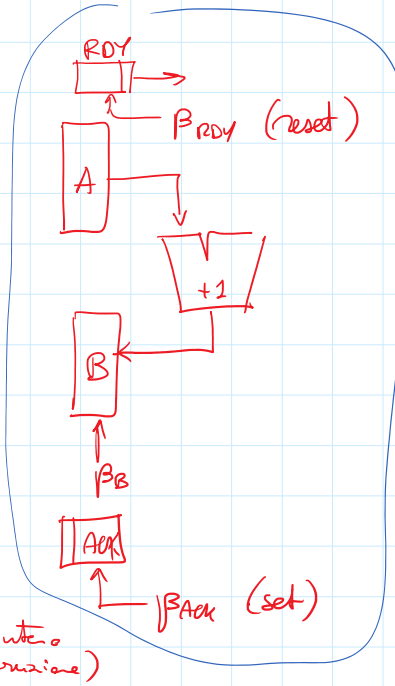




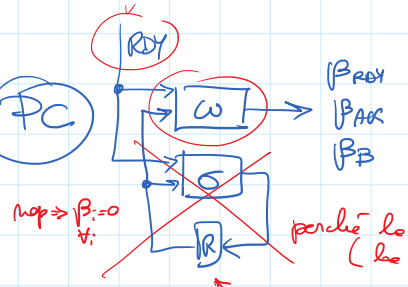


Ø. (RDY = Ø nop, Ø
(= 1) A+1 → B,
set ACK,
reset RDY,
Ø



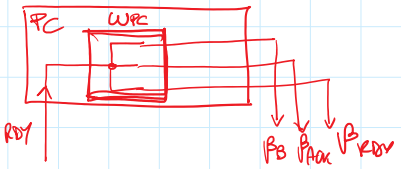


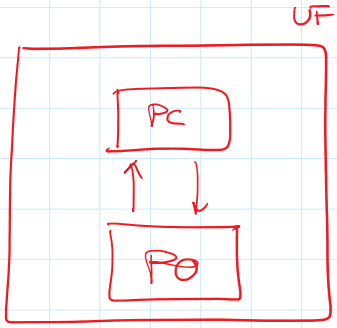
RDY	w_{PC}	β_{RDY}	β_{ACK}	β_B
0		0	0	0
1		1	1	1

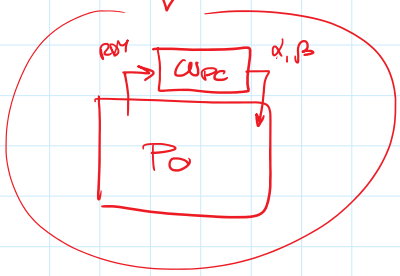


perché ho 1 unica stato interno (ho 1 unica μ-istruzione)

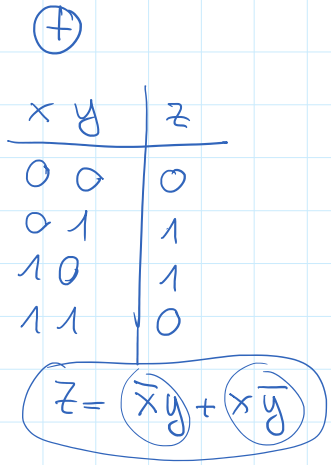
↖ rappresenta il # di μistruzione corrente

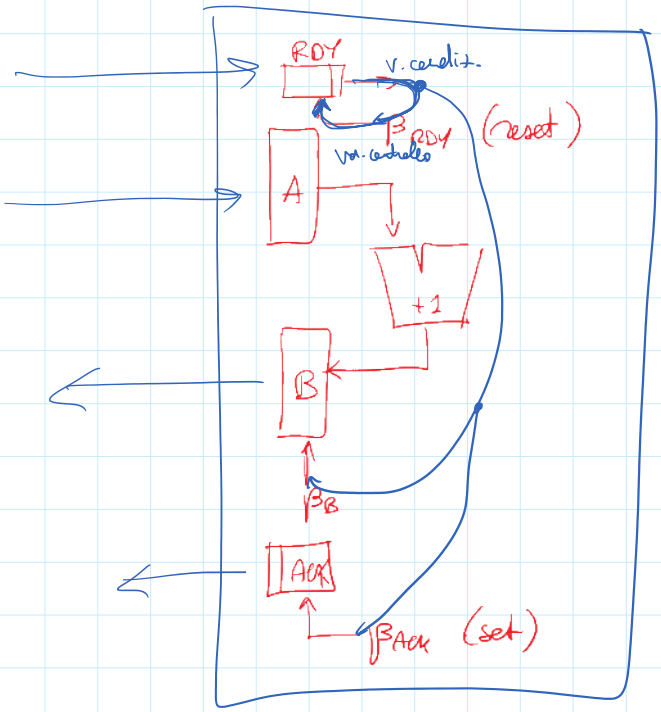


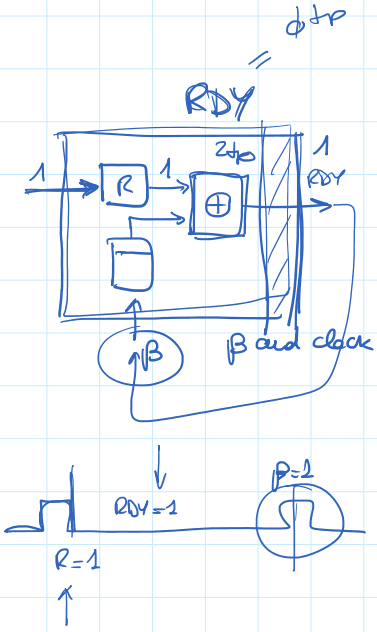




1 unica μ-istruzione
⇒ UF è fatta utilizzando 1 sola rete sequenziale


⊕
x y | z
0 0 | 0
0 1 | 1
1 0 | 1
1 1 | 0
z = x̄y + xȳ

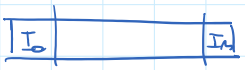

RDY
v. condiz.
β RDY
v. codello
(reset)
A
+1
B
βB
ACK
βACK (set)


ϕtp
RDY
1
R
1
2tp
1
RDY
β
β and clock

RDY=1
R=1
R=1

Varianti di condizionate

Semplici $\not\exists t_p$ { I_0, I_n, RDY, ACK }



Complesse $\exists t_p$ $k \neq \phi$

- o accessi alla memoria $M[I]_0$
- o operazioni con ALU o reti combinatorie particolari

$OR(A)$ or bit a bit dei bit di A

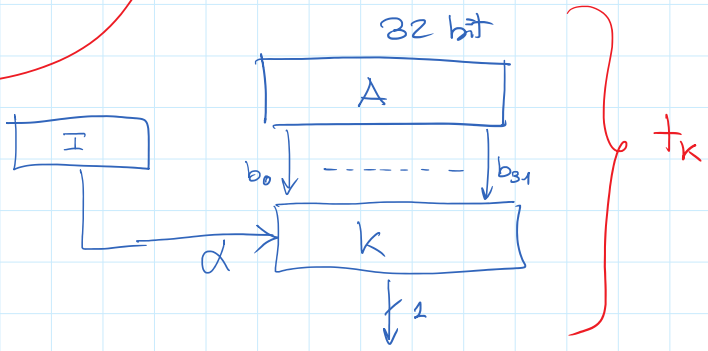
Segno $(A-B)$

Segno $(M[I]_\phi - B)$

$(A_I = 1)$ (reg, reg)

Semplice?

Complessa?



8

mercoledì 11 ottobre 2017 09:12

10

mercoledì 11 ottobre 2017 09:12

11

mercoledì 11 ottobre 2017 09:12

12

mercoledì 11 ottobre 2017 09:12