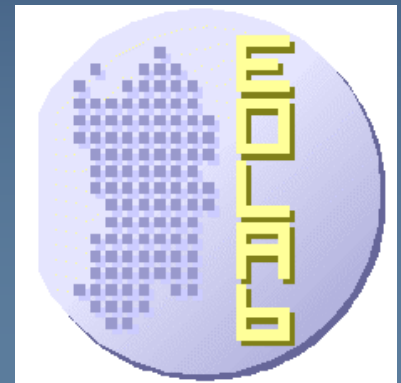


Architetture Network on Chip



Paolo Meloni

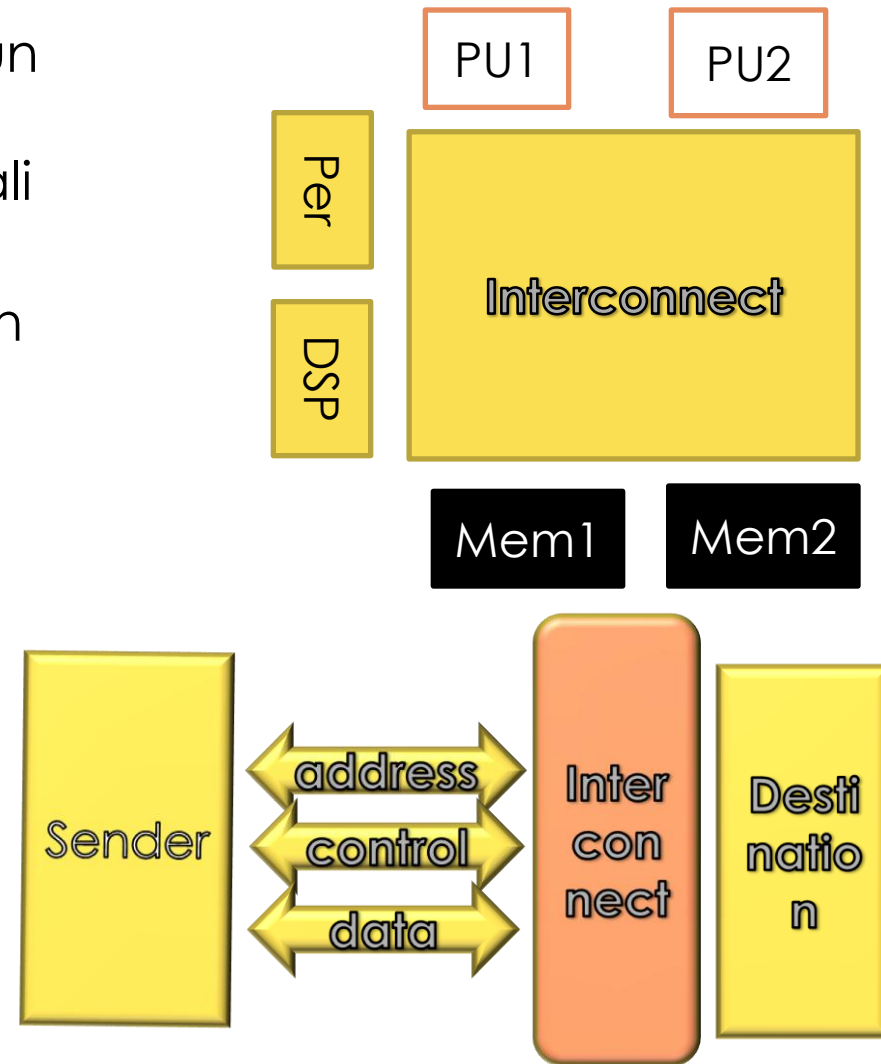
Universita' degli studi di Cagliari



Sistemi di interconnessione on-chip



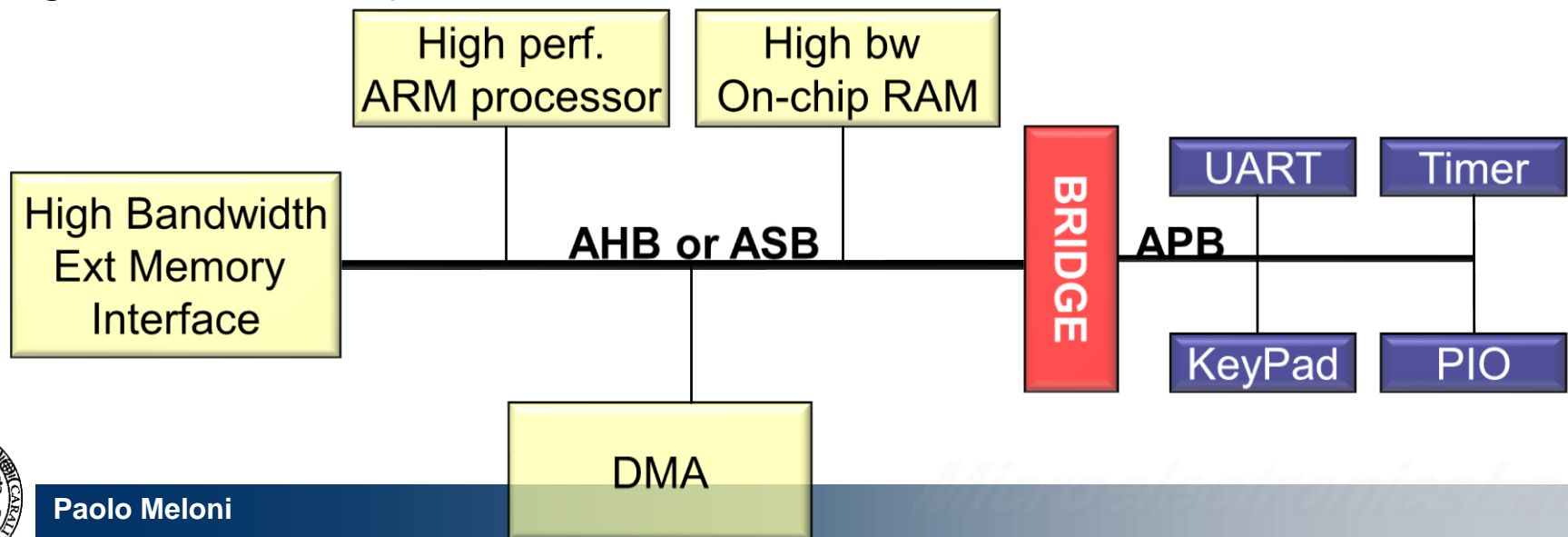
- Sono le strutture che, all'interno di un sistema on-chip, permettono la comunicazione tra le unità funzionali
- Ogni IP CORE connesso al sistema comunica con l'esterno inviando un certo numero di informazioni per ogni operazione.
- Dal punto di vista del sistema di comunicazione ogni scambio di informazioni tra core è vista come una transazione
- trasferimento di
 - un indirizzo
 - un dato
 - un certo numero di informazioni di controllo



Sistemi di interconnessione shared bus



- I sistemi di interconnessione più semplici sono quelli di tipo shared bus
- Tutti i core sono connessi a una struttura di comunicazione condivisa (tipicamente un fascio di wire), che può essere pilotata alternativamente dai vari core ad essa connessi. Tipicamente le informazioni sul bus sono trasmesse in broadcast.
- Allo stato dell'arte vengono utilizzate strutture più complesse in cui elementi shared sono interconnessi tra loro per ottenere un miglioramento in performance.





- Nascono come una soluzione alternativa ai sistemi di tipo shared bus.
- Tali sistemi mostrano infatti problemi di scalabilità a livello
 - Funzionale
 - tecnologico

Vengono mutate a livello on-chip le tecniche di comunicazione packet-switched già studiate nel campo delle interconnessioni tra computer



Scalabilità funzionale

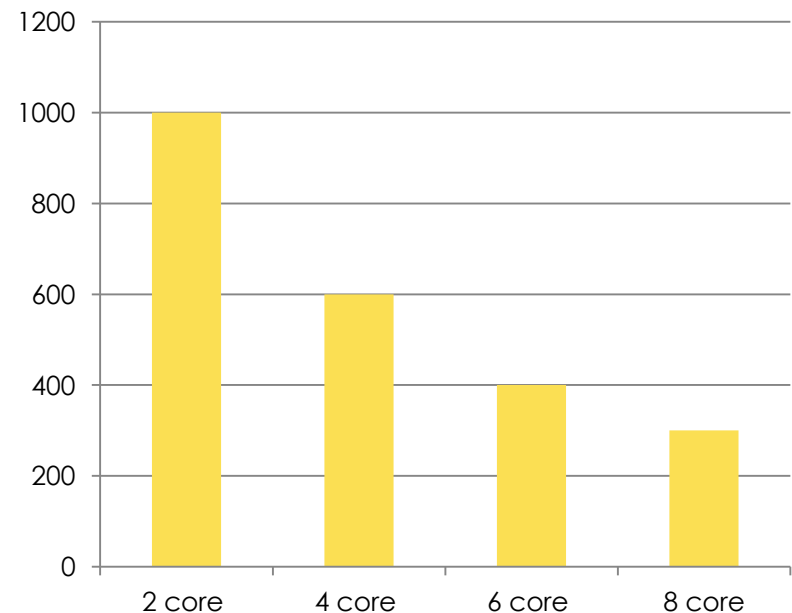


- I sistemi di tipo shared bus sono strutture di tipo condiviso.
- Le unità connesse non possono accedere contemporaneamente alla risorsa di comunicazione.
- La banda di comunicazione disponibile per ogni core diminuisce con l'aumento dei core connessi alla rete

•Esempio:

- Bus condiviso - 32 Bit
- Frequenza di funzionamento=100 MHz
- Traffico sostenibile dal bus= $32 \text{ Bit} * 100 * 10^6 * \text{sec}^{-1} = 3,2 \text{ Gbit/sec}$
- Numero di IP core connessi = 4
- Traffico iniettabile da ogni core (av.)= $0,8 \text{ Gbit/sec}$
- Numero di IP core connessi = 8
- Traffico iniettabile da ogni core (av.)= $0,4 \text{ Gbit/sec}$

banda disponibile per core



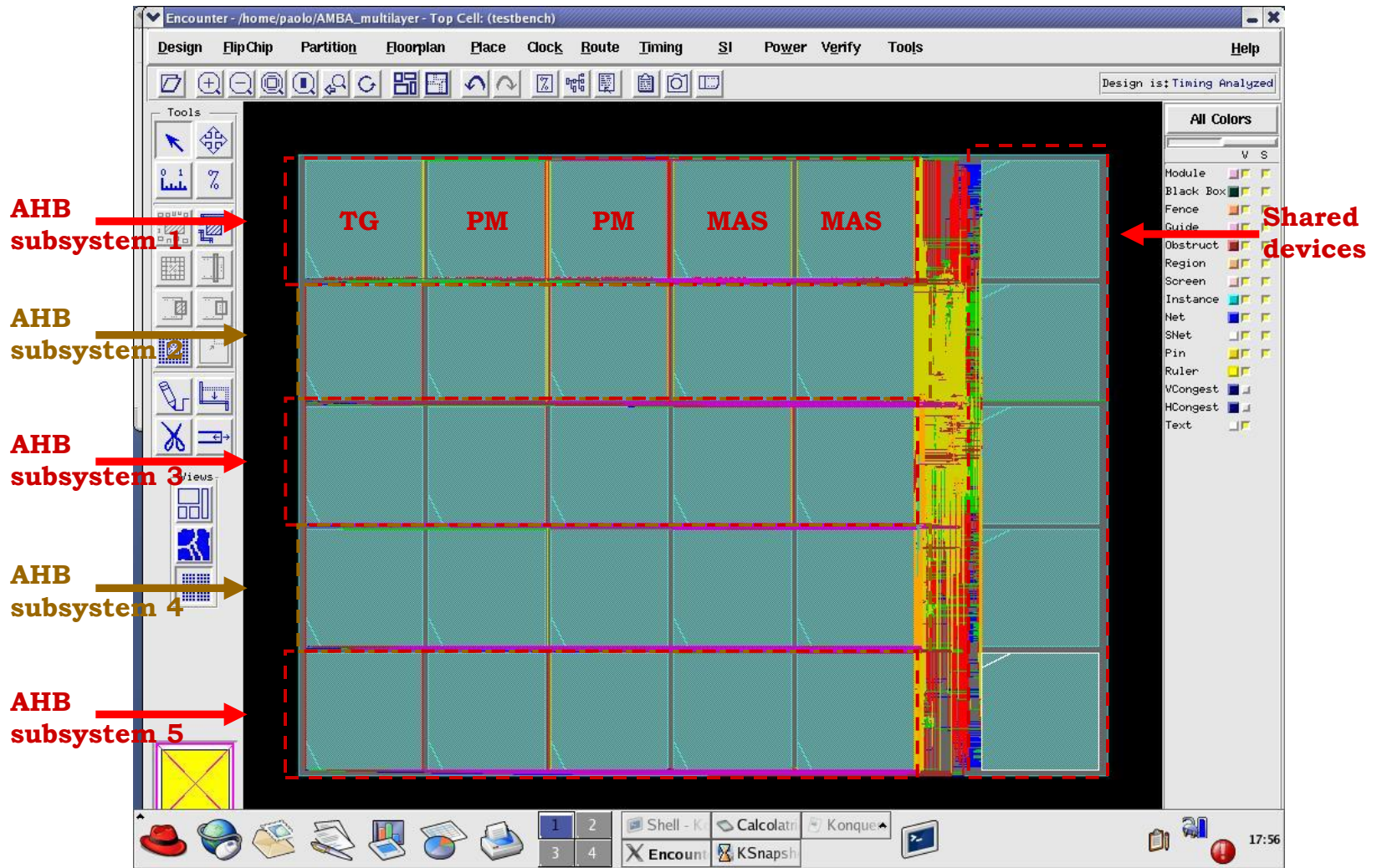
Scalabilità – aspetto tecnologico



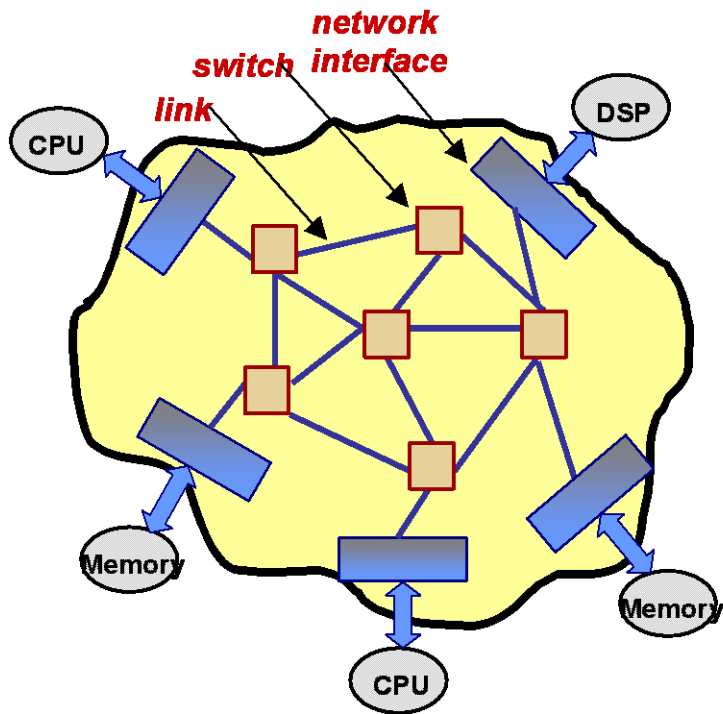
- Con lo scaling dei processi tecnologici
 - Il ritardo relativo alle porte logiche diminuisce
 - Il ritardo relativo ai wire di interconnessione rimane costante o aumenta
- In fase di implementazione su silicio, la gestione dei lunghi wire tipici dei sistemi bus-based risulta critica e limitante per le performance in termini di
 - Delay : il ritardo legato alle interconnessioni influisce in maniera significativa sulla massima frequenza di funzionamento dell'intero sistema.
 - Power: i wire sono caratterizzati da un valore di capacità non più trascurabile, quindi le celle che si occupano di pilotare il valore di tali wire saranno caratterizzate da un elevato consumo di potenza.



Shared bus system layout



Architetture Network on Chip



- Una architettura NoC è formata da diverse istanze di tre blocchi fondamentali:
- **Network interface:** si occupa di operare la conversione per adattare il protocollo di comunicazione tipico dei core al protocollo a pacchetti tipico della rete NoC
- **Switch:** si occupa delle operazioni di routing.
- **Link:** è la struttura fisica (tipicamente un fascio di wire ma può contenere anche dei repeater) che collega gli switch tra loro e con le NI

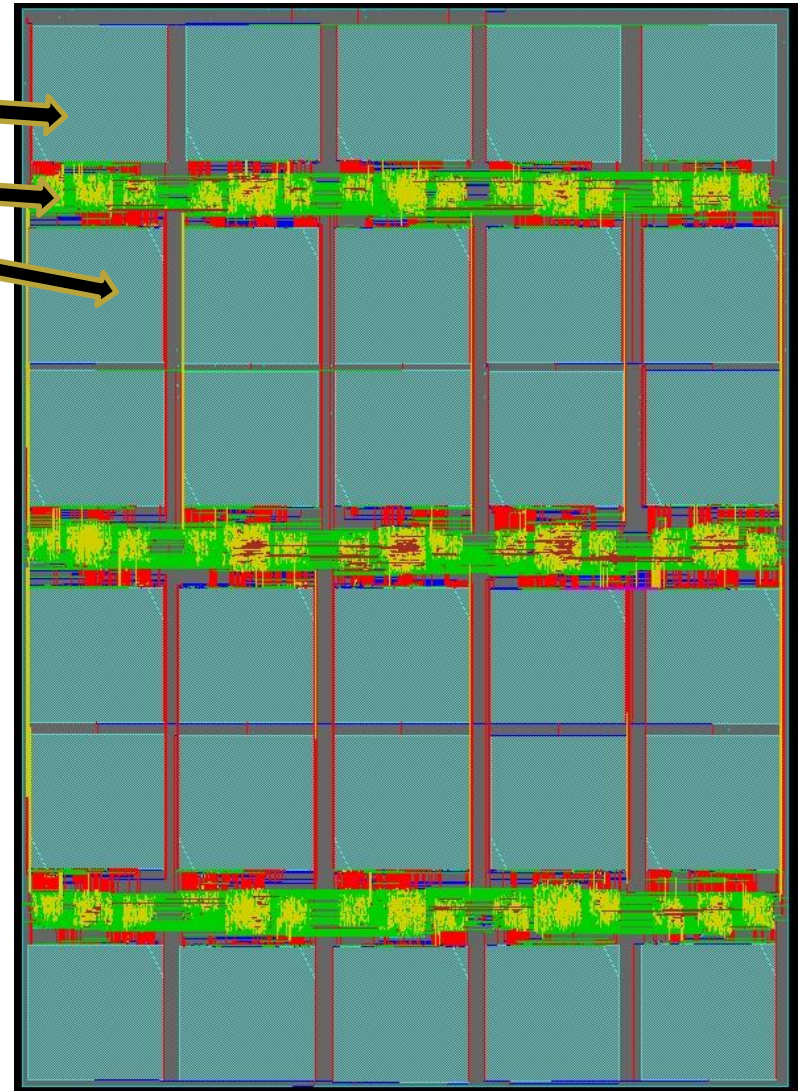
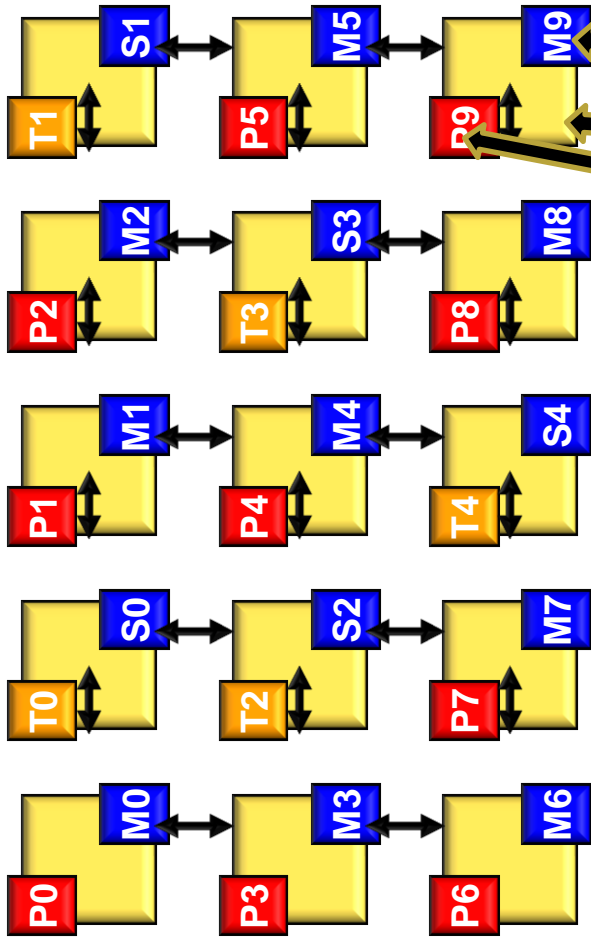




- I NoC forniscono alcune feature che permettono di sopperire alle carenze di scalabilità dei sistemi bus-based
- Tra queste:
 - Supporto contemporaneo di transazioni multiple
 - Segmenti al posto di linee globali:
 - Comunicazione a pacchetti
- Disponibilità di diversi gradi di libertà per il progettista:
 - topologia
 - routing
 - ampiezze dei bus
 - numero di porte
 - controllo di flusso



Network on chip layout



Concetti base



- Topologia
- Unità di informazione
- Algoritmi di routing
- Implementazione





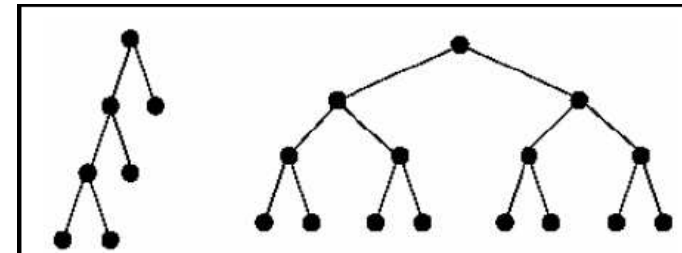
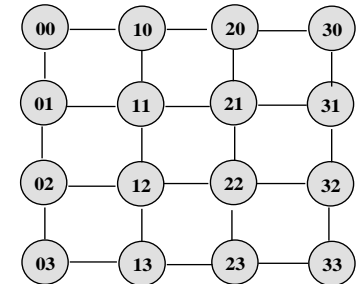
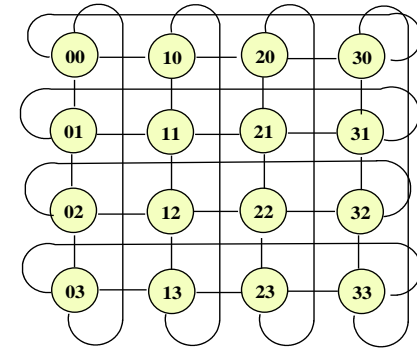
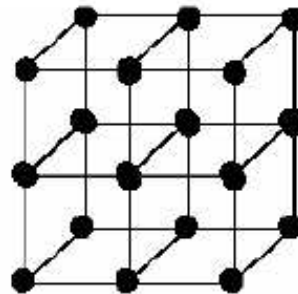
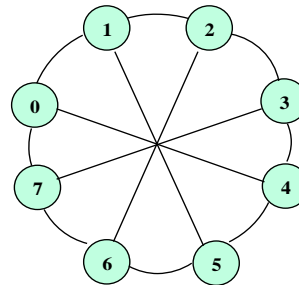
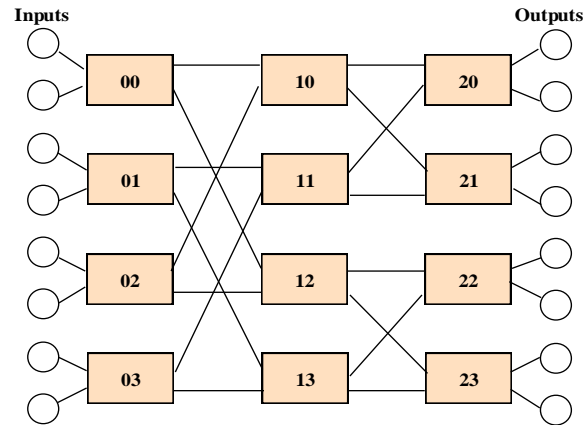
- **Messaggio**: unità logica di informazione di dimensione maggiore (solitamente coincide con tutto l'insieme di informazione inclusa nella transazione).
- **Pacchetto**: unità logica minima trasferibile attraverso il protocollo di rete.
- **Flit**: (flow control unit) unità logica trasmissibile dal punto di vista della policy relativa al controllo del flusso.
- **Phit**: (physical unit) unità logica fisicamente trasmissibile attraverso il link. Spesso coincide col flit.



Topologia



- La topologia identifica la maniera in cui i nodi della rete (gli switch) sono interconnessi tra loro
- Butterfly
- Torus
- Spidergon
- Mesh
- Hypercube
- Balanced or unbalanced tree
- **Un ruolo molto importante è rivestito dalle topologie irregolari: possono essere configurate in maniera ottimale in base alla applicazione in uso**



Esistono diverse classificazioni degli algoritmi di routing:

a) In base alle informazioni di cui si tiene conto durante la generazione del percorso di routing

- **Routing deterministico:** la generazione del percorso di routing avviene solo in base alla conoscenza di informazioni riguardanti l'indirizzo del sorgente e l'indirizzo del destinatario, trascurando qualunque informazione relativa allo stato in cui versa la rete.

- **Routing semi-deterministico:** la coppia sorgente destinatario, a differenza di quanto accade nel routing deterministico, non identifica in maniera univoca un percorso, ma più percorsi diversi, da scegliere casualmente o in maniera ciclica.

- **Routing adattativo:** vengono utilizzate le informazioni relative al traffico ed allo stato della rete per determinare il percorso di routing, allo scopo di non incappare in congestioni o guasti già verificati.



b) In base al punto della rete in cui vengono prese le decisioni riguardo al percorso di routing

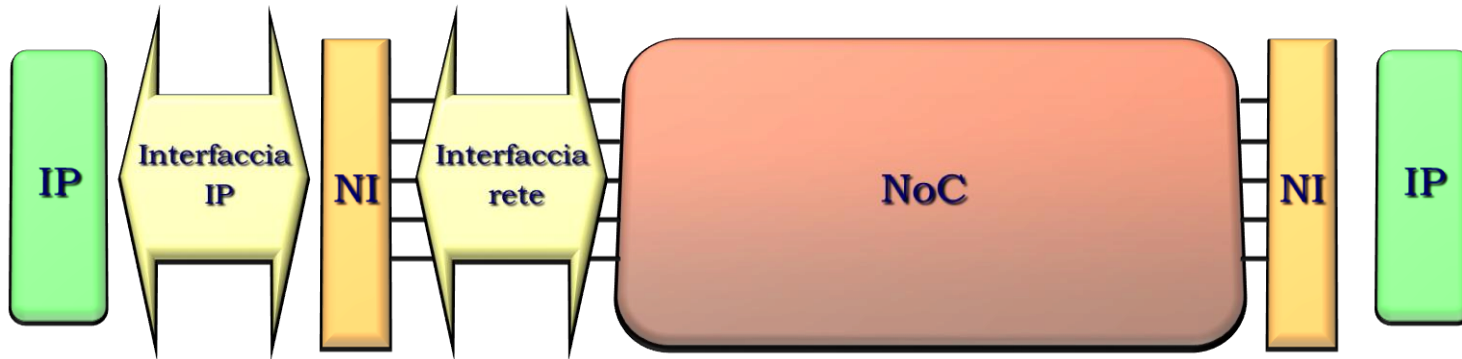
- **source routing:** il percorso che dovrà seguire un pacchetto è determinato al nodo sorgente e il compito dei diversi switch è solo quello di leggere nell'header di ogni pacchetto in quale uscita devono essere smistati i diversi pacchetti.
 - K uscite per ogni switch e N switch da attraversare
 - numero di bit richiesti per la codifica del path = $N \log_2 K$.
- **routing distribuito:** la funzione di routing è calcolata in ogni switch per ogni pacchetto che lo attraversa. L'header contiene solo l'informazione dell'indirizzo della destinazione finale: spetta allo switch stesso scegliere verso quale uscita indirizzarlo.
- **routing ibrido:** a monte si calcolano solo alcune destinazioni parziali, più ovviamente quella finale. Gli switch provvedono nel dare la giusta direzione ai pacchetti che transitano in zone intermedie;
- **routing centralizzato:** in questo caso la funzione di routing è stabilita da un controllore centralizzato.



Blocchi base: Network Interface



- La network interface si occupa della codifica delle informazioni che un core intende trasferire secondo il protocollo di rete.
- Il core comunica verso l'esterno tramite un suo particolare tipo di interfaccia (tipicamente di tipo bus-based)
- La NI converte la transazione iniziata dal core in pacchetti



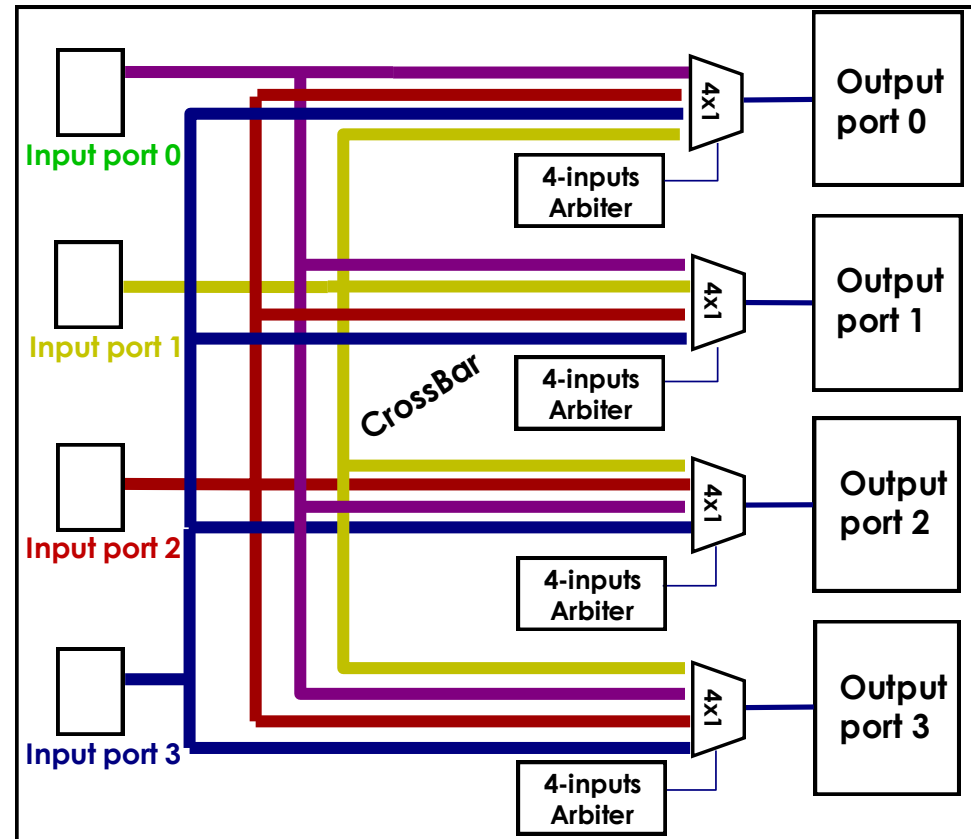
- Esistono in generale due tipi di Network Interface:
 - NI master : interfaccia con un core di tipo slave
 - NI slave : interfaccia con un core di tipo master



Blocchi base: Switch



- Lo switch si occupa di istradare i flit attraverso la rete.
- Ha un certo numero di porte di ingresso e un certo numero di porte di uscita
- Generalmente contiene le risorse di buffering che servono per ridurre la congestione e per evitare perdite d'informazione
- A seconda del tipo di politica di routing scelto deve decodificare o meno l'indirizzo destinazione ed analizzare o meno ogni flit





- **Circuit Switching** : il primo flit del primo messaggio, contenente le informazioni di routing, viene inviato alla rete e ogni volta che il flit attraversa uno switch prenota il percorso. Un segnale di acknowledgment viene trasmesso alla sorgente e quando viene ricevuto, questa spedisce a piena banda tutti gli altri flit lungo il medesimo circuito, che viene liberato solo dopo la conclusione del messaggio.
- **Packet Switching**:
 - **Store-and-Forward** : tutti i flit di un pacchetto vengono pacchetti, che vengono memorizzati completamente in ogni switch prima di essere ritrasmessi a quello seguente.
 - **Wormhole Switching** : l'informazione riguardo al routing di tutto il pacchetto è contenuta nel primo flit. Lo switch analizza solo il primo flit e instrada tutti gli altri flit del pacchetto nella stessa direzione sino alla ricezione dell'ultimo flit. La bufferizzazione può avvenire attraverso stadi di pipeline nel link di collegamento fra switch.



Blocchi base - Link



- Sono i collegamenti fisici tra switch e switch e tra switch e NI
- Hanno una larghezza (in bit) che determina la massima quantità di informazione trasferibile per ogni colpo di clock (il phit). Di solito questa larghezza coincide anche col flit
- Possono contenere dei repeater che permettono di rigenerare il segnale o degli stadi di buffering che permettono di mantenere la frequenza di funzionamento della rete indipendentemente dalla lunghezza dei link



Implementazione – Xpipes NoC architecture

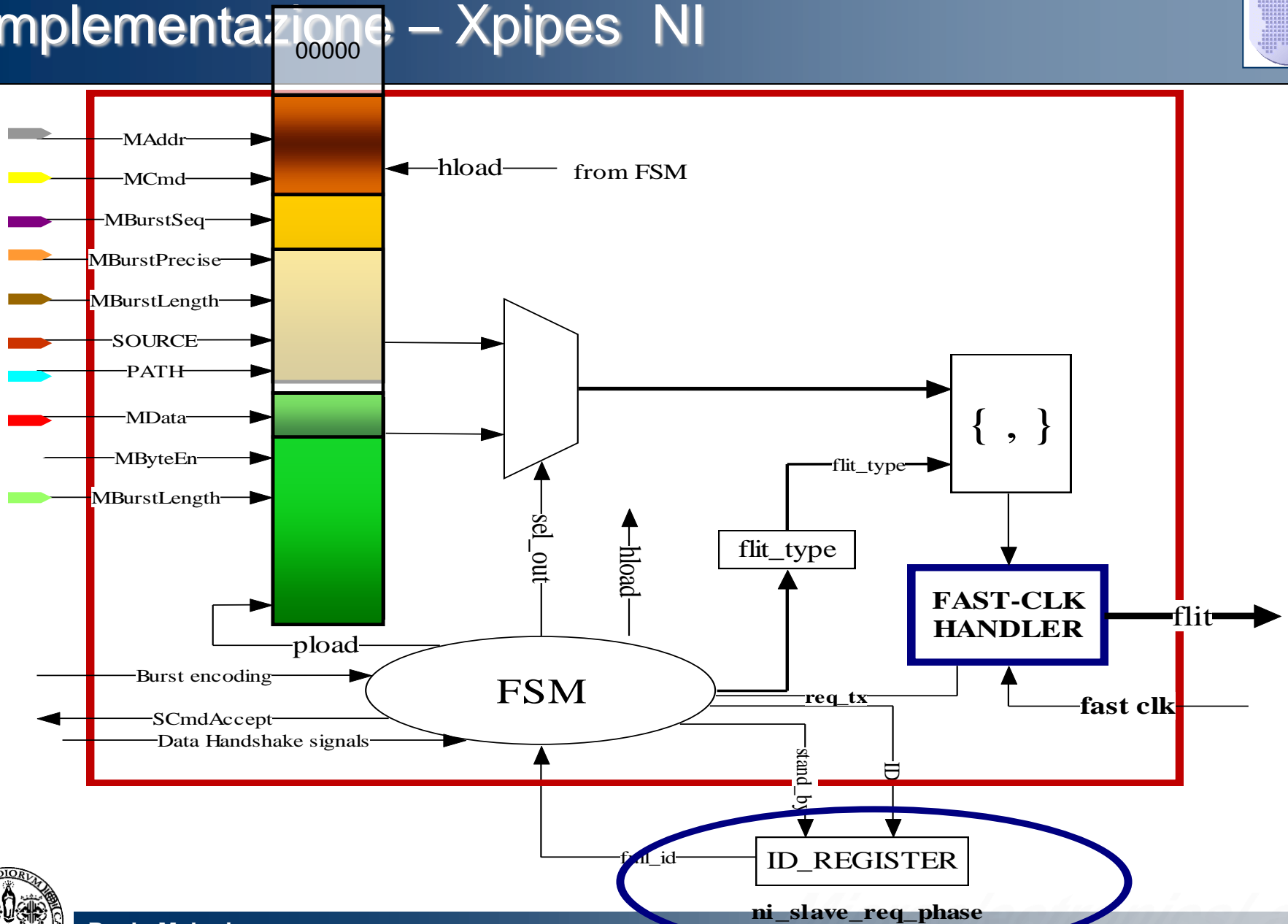


- Architettura sviluppata qui a Cagliari in collaborazione con
 - **Stanford University – Ca, USA**
 - **EPFL – Lausanne, Svizzera**
 - **Universtità di Bologna**

- **Features**
 - Topologia completamente configurabile
 - Supporto per le topologie irregolari
 - Blocchi completamente configurabili
 - Source routing deterministico
 - Wormhole swithching



Implementazione – Xpipes NI



Implementazione – Xpipes switch

