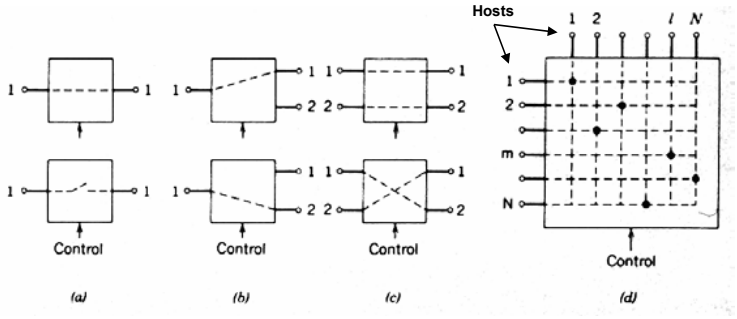
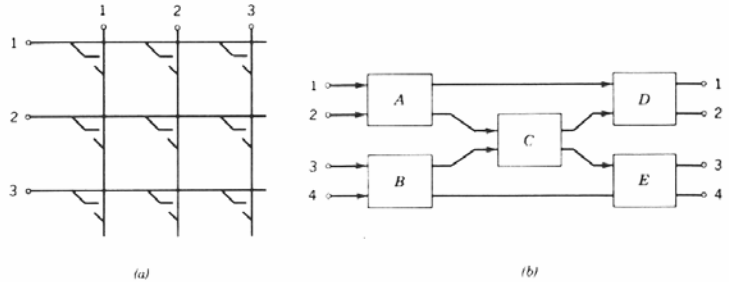
Data switches

Servono a interrompere (interruttore) o a commutare (commutatore) un flusso di dati tra due nodi (*hosts*)



a - 1 x 1 *(on-off)* serve a connettere o a disconnettere due linee
b - 1 x 2 connette una linea con una delle altre due
c - 2 x 2 *(crossbar)* connette due linee a due linee
d - N xN (*crossbar*) connette N linee a N linee

Nel caso *crossbar* (1) ogni linea d'ingresso può in ogni situazione essere connessa ad una linea d'uscita senza bloccaggio: ***non-blocking***. Ciò non è vero in generale per altre topologie, per esempio (2).

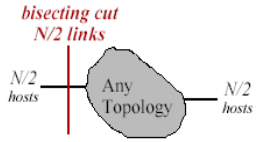


1 - *switch* 3 x 3 costruito con 9 *switches* 1 x 1
2 - *switch* 4 x 4 costruito con 5 *switches* 2 x 2

partition:
dividere la rete in due o più sottoinsiemi che non comunicano tra di loro

connectivity:
numero di vie possibili tra due processori (nodi, *hosts*), oppure minimo numero di connessioni che si devono togliere per dividere la rete in due sottoinsiemi indipendenti (*partition*)

bisettrice
(bisecting cut, bisection width):
divide la rete in due gruppi, ciascuno di N/2 processori (nodi, *hosts*)



Se i due sottoinsiemi contengono lo stesso numero di processori (nodi, *hosts*):
connectivity = bisecting cut ≡ bisection width

diametro medio:
numero di *links* tra processori/nodi percorsi dalla connessione più breve tra due processori scelti a caso

distanza media *(average distance):*
numero di *links (hops)* successivi verso una destinazione aleatoria

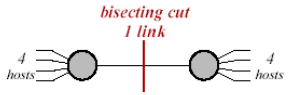
degree:
numero di *links* uscenti/entranti in un nodo

unicast
connessione (messaggio) da un processore/nodo verso **un** processore/nodo

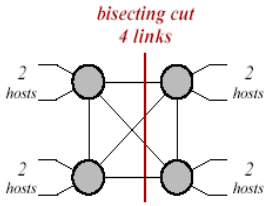
multicast:
connessione (messaggio) da un processore/nodo verso **più** di un processore/nodo

broadcast:
connessione (messaggio) da un processore/nodo verso **tutti** gli altri processori/nodi

bisettrice minima:
minimo numero di connessioni tra due gruppi, ciascuno di N/2 processori/nodi



bisettrice completa (*full bisection*):
quando il numero di connessioni tra due gruppi, ciascuno di N/2 processori/nodi, è uguale a N/2



diametro massimo
numero massimo di *links* tra processori/nodi percorsi dalla connessione più breve tra due processori/nodi: è una misura della latenza massima.

rearrangeable
se la rete (*switch*) è capace di connettere ogni permutazione di connessioni tra ingressi e uscite, con eventuale scelta della migliore configurazione.

Per esempio, uno switch 8x8 deve essere capace di stabilire le connessioni

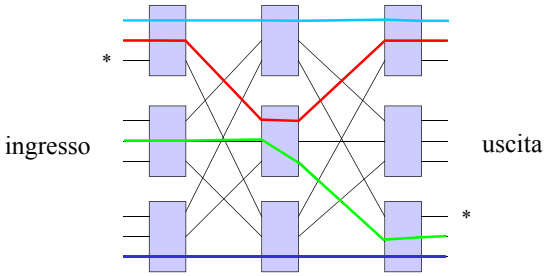
dal processore	0	1	2	3	4	5	6	7
	|	|	|	|	|	|	|	|
al processore	3	5	1	0	7	6	2	4

e tutte le rimanenti 8! - 1 = 40319 permutazioni possibili.

Una rete *rearrangeable* possiede bisettrice completa, ma una rete con bisettrice completa non è necessariamente *rearrangeable*.

self-routing
se lo *switch* elementare interpreta al passaggio l'informazione di destinazione contenuta nel pacchetto.

blocking interno, nello *switch*
quando non si può stabilire una connessione tra un ingresso e una uscita entrambi liberi

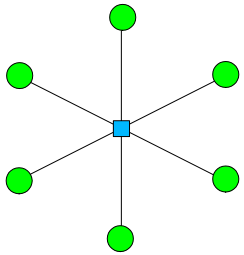


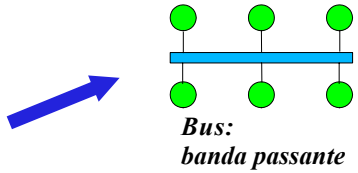
Head Of the Line **(HOL)** e ***Output blocking***
code in ingresso o in uscita

speedup
rapporto tra la velocità di trasferimento dati dall'ingresso all'uscita dello *switch* e la velocità di linea in ingresso

Topologia

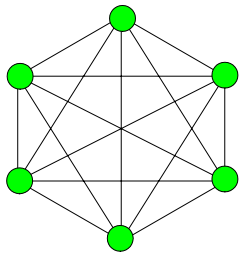
Connessione completa, senza *switches*



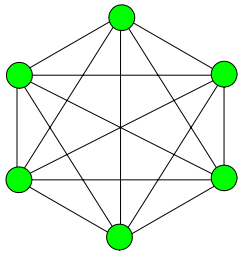


Bus:
banda passante $\propto N$

N numero di nodi (*hosts*)



Switches in ogni nodo
(commutazione distribuita)
e **connessione completa**



degree: numero di connessioni ad ogni nodo: $N-1$

diametro massimo: numero massimo di *links* nella connessione piu breve tra due nodi: **1**

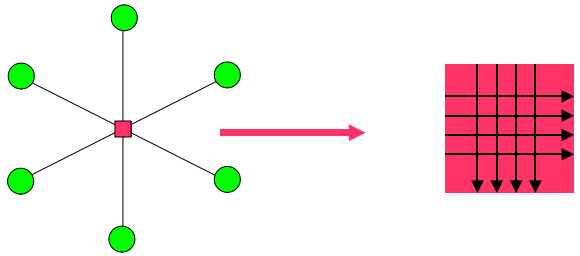
distanza media: 1

***bisection width*:** $(N/2)^2$

Switches in ogni nodo, diverse topologie

	Degree	Max_Diam	Ave Dist	Bisection
1D mesh	≤ 2	N-1	N/3	1
2D mesh	≤ 4	$2(N^{1/2} - 1)$	$2N^{1/2} / 3$	$N^{1/2}$
Ring	2	N / 2	N/4	2
2D Torus	4	$N^{1/2}$	$N^{1/2} / 2$	$2N^{1/2}$
n-Hypercube	n	n = LogN	n/2	N/2

Switch centrale ==> *Crossbar Switch*



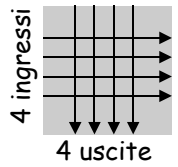
È concettualente semplice aumentare le dimensioni N x N di un *crossbar switch:*

- *non-blocking*
- il numero di elementi e la potenza salgono come $O(N^2)$

È possibile costruire l'equivalente di un grande *crossbar switch* con elementi connessi in stadi successivi? Rete di piccoli *crossbar switches* o altra topologia?

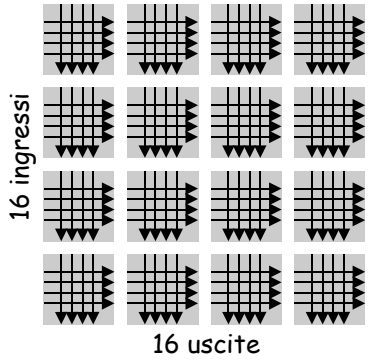
Crossbar switch costruito con circuiti elementari

circuito elementare:

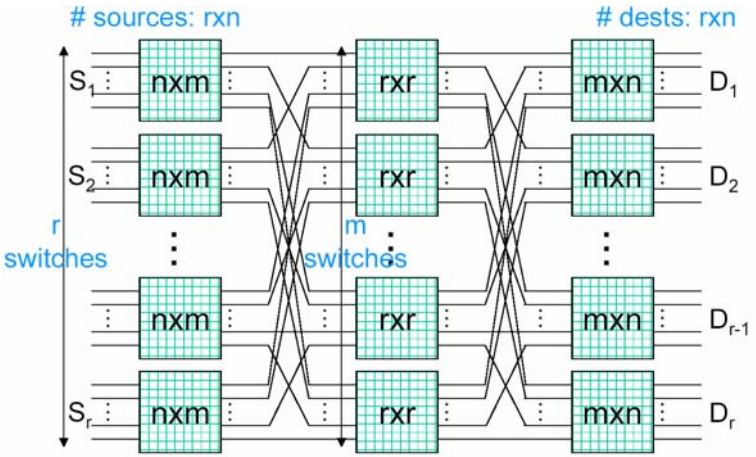


Si devono trasmettere ingressi e uscite all'elemento successivo!

16x16 *crossbar switch*:



Clos Network (1)



non-blocking quando $m \geq 2n-1$

Clos network ==> tre stadi

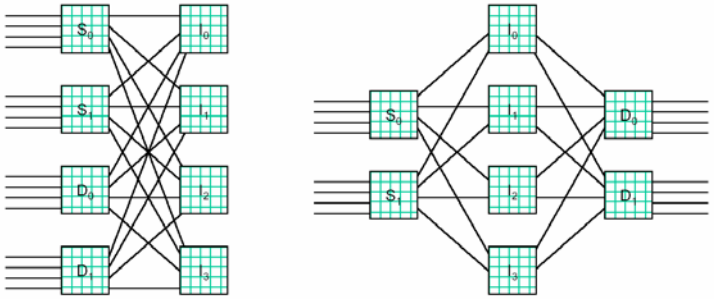
rn ingressi e *rn* uscite

gli **stadi di ingresso e uscita** (*leaves*) sono uguali
- hanno *r* elementi asimmetrici *n* per *m*
- permettono la connessione di *rn* nodi (*hosts*) in ingresso e *rn* in uscita

lo **stadio intermedio (*spine*)**
- ha *m* elementi simmetrici *r* per *r*
- la sua dimensione determina le proprietà di bloccaggio

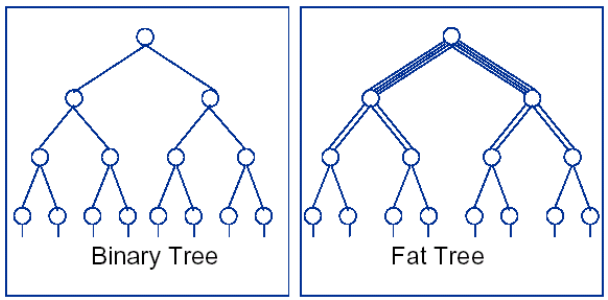
non-blocking quando $m \geq 2n-1$ per connessini *point-to-point*

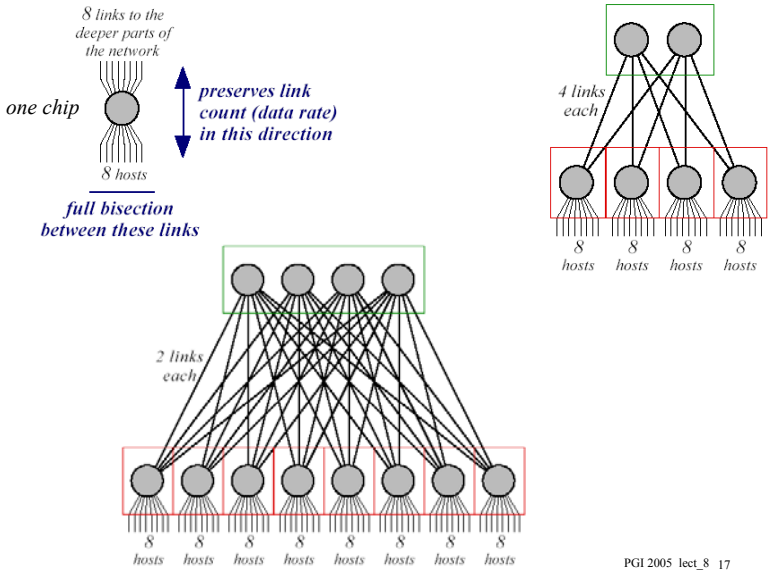
Clos Network (2)

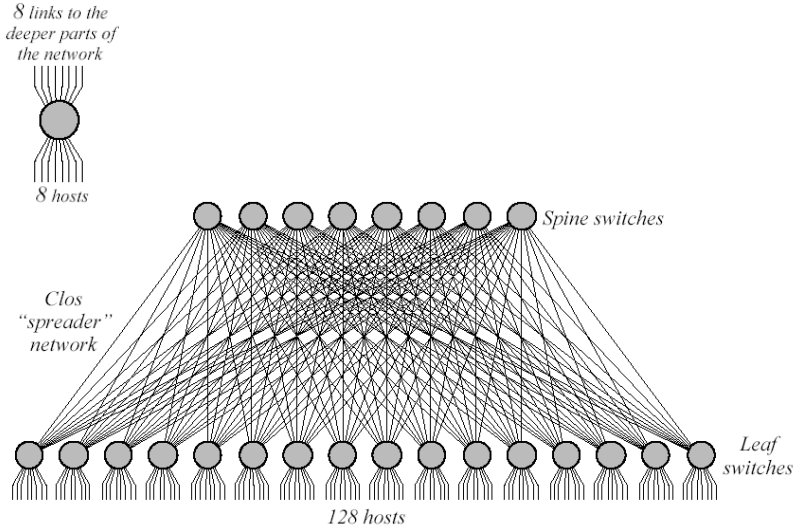


Quando i links sono bidirezionali non c'è distinzione tra ingressi e uscite

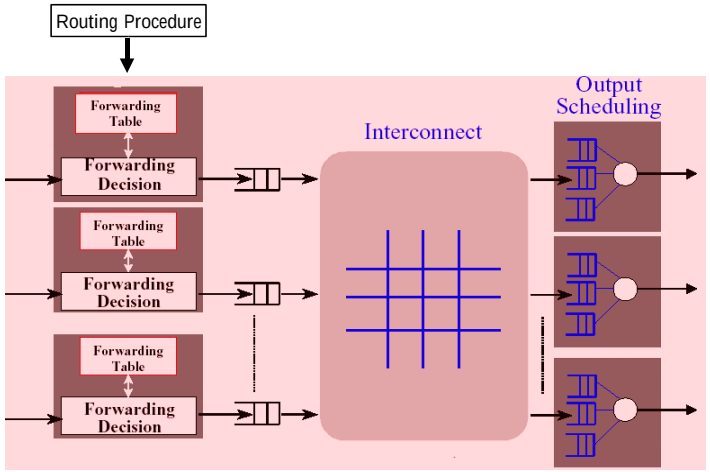
Le reti di Clos sono anche chiamate
Constant Bisectional Bandwidth (**CBB**) oppure ***Fat Tree***







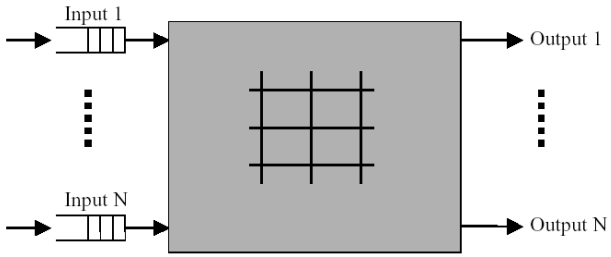
Routing algorithms



Routing Algorithms

Sistema	a commutazione di circuiti (*circuit switching*), es. telefono a commutazione di pacchetti (*packet switching*), es. rete locale
Protocollo	*connection oriented*, es.telnet *connectionless*, es TCP/IP
Percorso (*routing*)	precalcolato *offline*, *look-up-table* precalcolato al momento contenuto nel pacchetto
Controllo	centrale, usa un percorso precalcolato locale, legge l'indirizzo del nodo seguente contenuto nel pacchetto

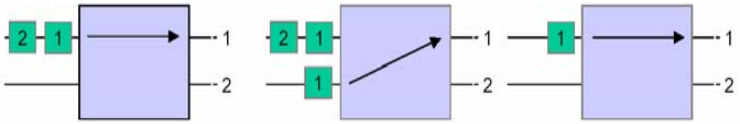
Input Queuing (IQ)



- Una coda per ingresso
- Di solito coda FIFO
- La coda accumula i pacchetti che non possono partire subito
- Appena un pacchetto arriva alla porta di uscita parte immediatamente (non ci sono code in uscita)
- Funziona con *switches* lenti, *speedup* = 1
- *Head-Of-Line (HOL) blocking*

Head of Line (HOL) Blocking

Problema intrinseco alla natura della FIFO in IQ



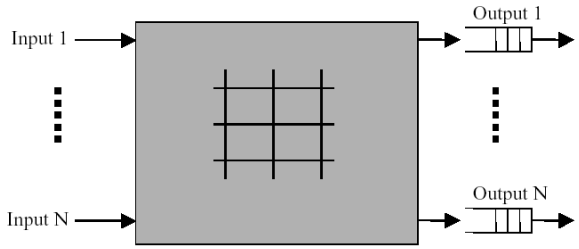
il paccketto per la porta 2 è bloccato dal pacchetto per la porta 1

il paccketto per la porta 2 è bloccato dal pacchetto in attesa della porta 1

non ci sono paccketti per la porta 2

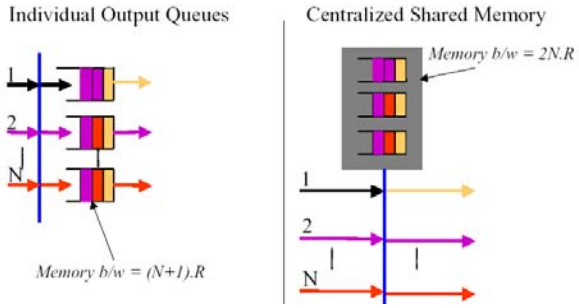
In ciascuno dei tre casi la porta 2 in uscita non è utilizzata.
Con distribuzione aleatoria dei pacchetti in arrivo, l'utilizzazione è inferiore al 58.6%

Output Queuing (OQ)



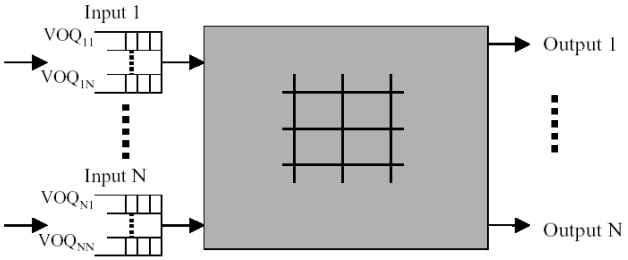
- Una coda per ogni uscita.
- Appena un pacchetto entra nello *switch* passa subito alla coda di uscita.
- La coda accumula i pacchetti che non possono partire immediatamente.
- Massimo *throughput*.
- Minima latenza
- Necessario uno *speedup* di *N* (numero di uscite).
- Non realizzabile per switches veloci con un numero elevato di porte

Output Queuing (OQ)



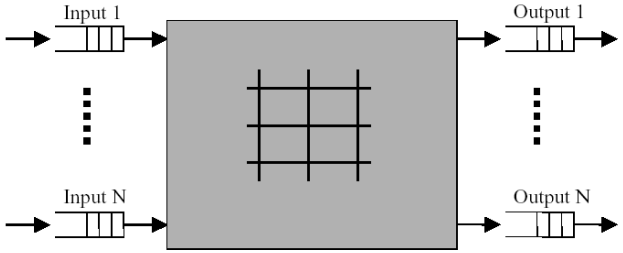
La banda passante delle code o della memoria deve essere sufficiente per ricevere tutti i messaggi diretti alla stessa destinazione; altrimenti *Block on Output Port*

Virtual Output Queuing (VOQ)



- I pacchetti diretti ad un ingresso si dividono in *N* code, una coda per ogni uscita (*N*x*N* code in totale).
- *Speedup* = 1.
- Evita *HOL blocking*.
- Algoritmi di priorità tra le code sullo stesso ingresso (vedi anche *barrel shifter*).

Combined Input Output Queuing (CIOQ)



- Code tanto in ingresso quanto in uscita.
- Lo scopo e di ottenere i vantaggi di IQ e OQ.
- algoritmi di priorità per le code
- Per uno *speedup* 1<S<N sono necessari *buffers* tanto in ingresso quanto in uscita per usare il sistema in modo efficiente
- Uno *switch* CIOQ con uno *speedup* di 2 si comporta come uno *switch* OQ con code FIFO, per ogni condizione (ragionevole) di traffico.

Parallel Packet Switching (PPS)

Costruire uno *switch* più veloce usando *K switches* lenti

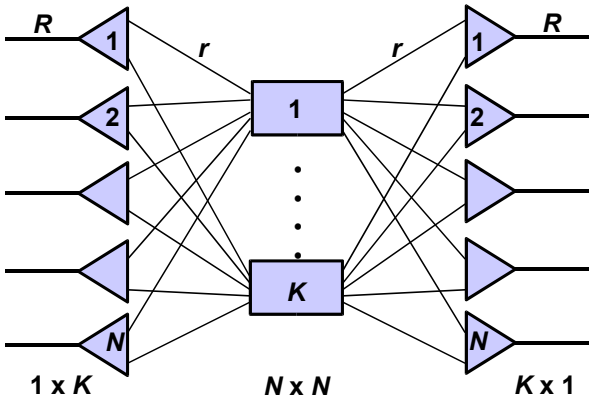
Topologia di una rete di Clos

N ingressi: *N demultiplexers* da 1 a *K*,
velocità d'ingresso *R*, d'uscita *r* = *R/K,*

Stadio intermedio: *K switches OQ* *N* x *N* ,
(*spine*)

N uscite: *N multiplexers* da *K* a 1,
velocità d'ingresso *r*, d'uscita *R*

Parallel Packet Switching (PPS)



Tecnologie di *switching*

Elettronica

Tanto i dati quanto il controllo sono segnali elettrici
Tradizionale in telecomunicazioni e reti di dati

Elettro-ottica

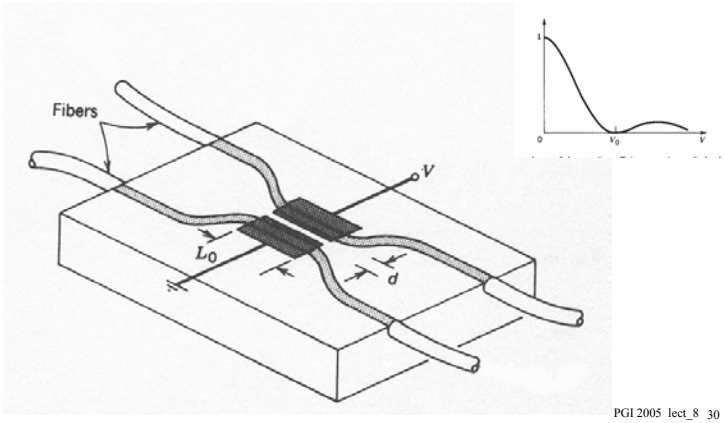
Il controllo è elettronico mentre i dati sono ottici
Directional couplers
MicroElectroMechanical mirror arrayS (MEMS)
In rapida espansione tanto in telecomunicazioni quanto in reti di dati

Ottica-ottica

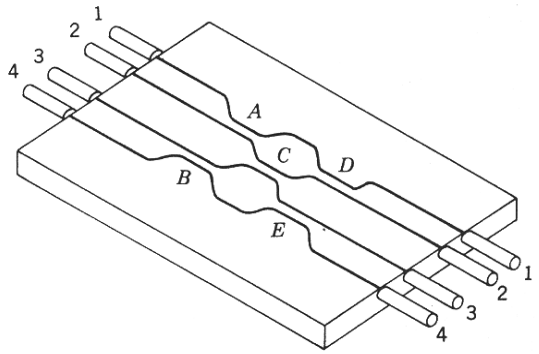
Tanto i dati quanto il controllo sono segnali ottici
Richiede tecniche elaborate, per esempio polarizzazione e ottica non lineare

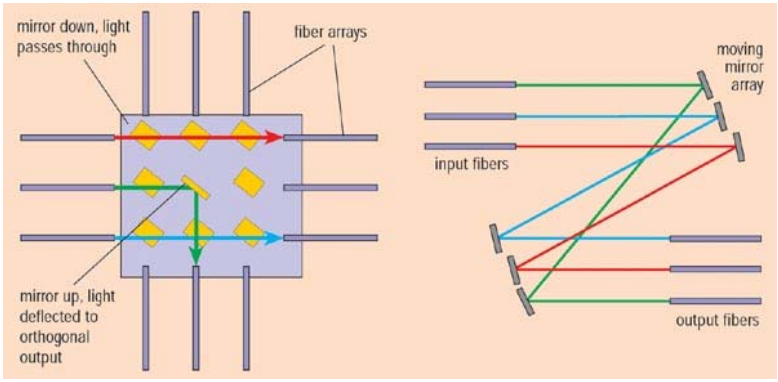
Switch realizzato con *directional couplers*

La tensione applicata ***V*** cambia l'accopiamento tra le due guide di luce

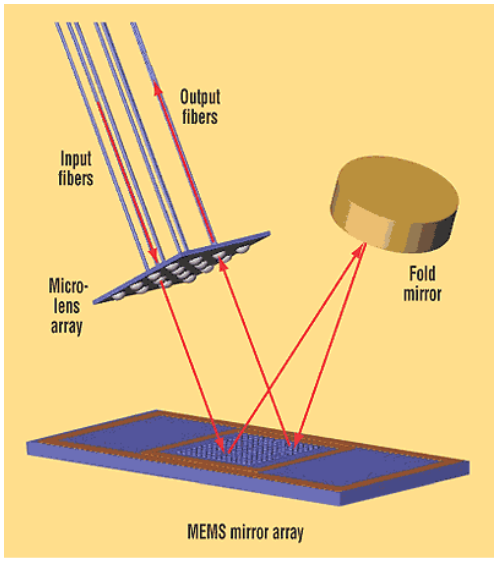


Con 5 *directional couplers A, B, C, D* ed *E* si realizza uno *switch* 4 x 4





2D MEMS 3D MEMS



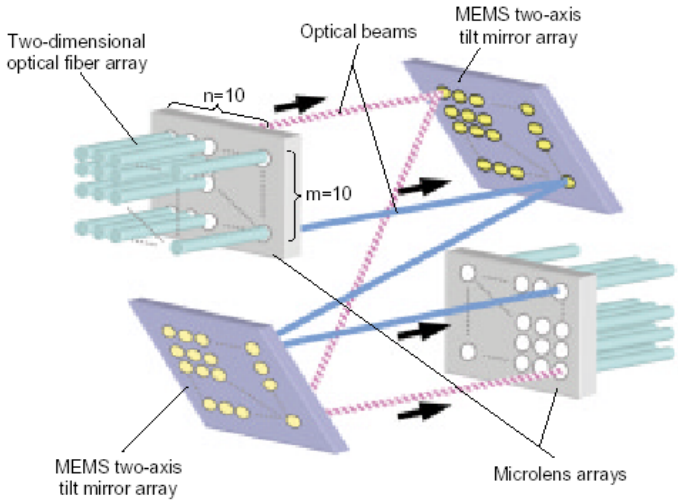


Fig. 1. Basic structure of 3D MEMS optical switch.

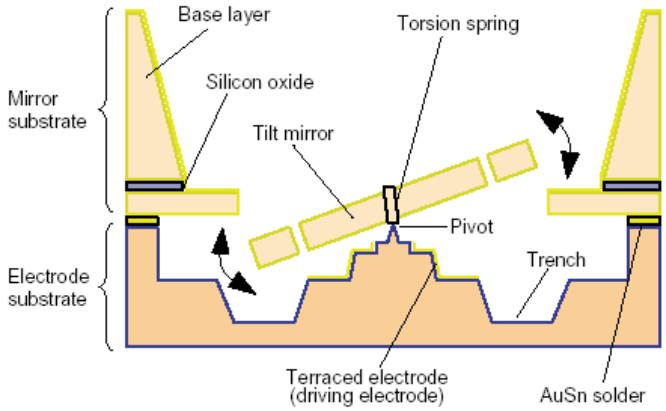


Fig. 2. Cross-sectional schematic of MEMS two-axis tilt mirror.

- **Caratteristiche di uno *switch***
- Dimensioni: numero di linee in ingresso e in uscita ovvero numero di connessioni, bidirezionali o no
- Tempo di commutazione, da 10 ps a 10 ms (MEMS)
- Banda passante per connessione e attraverso una bisettrice
- Tempo di propagazione
- Flusso, portata, *throughput*
- *Buffers* in ingresso e in uscita?
- Energia di commutazione 10 fJ – 20 fJ
- Dissipazione 1 µW per connessione
- *Insertion loss* *2±1*dB *(MEMS)*
- *Crosstalk*
- Dimensioni 1 cm^2 per 16 x 16 porte
- Riconosce o non riconosce gli indirizzi dei pacchetti?
  self-routing o no
- Tipo di protocollo:
  <u>connectionless</u> o <u>connection oriented</u>
  elettronico: può essere *self routing* e *connectionless*
  elettro-ottico: *connection oriented*

Referenze

Per una descrizione delle tecnologie elettroniche attuali di *switching*:

Guide to Myrinet-2000 Switches and Switch Networks, Myricom, http://www.myri.com/myrinet/m3switch/guide/

InfiniBand Clustering, Mellonanox Technologies, http://www.mellanox.com/technology/shared/IB_vs_Ethernet_Clustering_WP_100.pdf