Algoritmi per la Visualizzazione

Prof.ssa Tiziana Calamoneri

LAYOUT DI TOPOLOGIE DI INTERCONNESSIONE

Il modello di Thompson

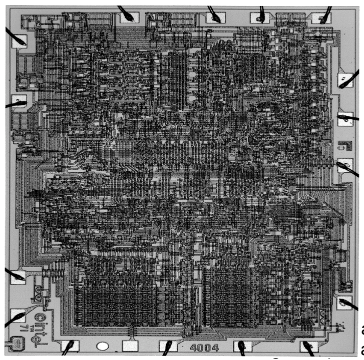
Il modello di Thompson (1)

- Il *problema del layout* di topologie di interconnessione nasce dal problema di stampare circuiti VLSI (Very Large Scale Integration) su un supporto di silicio.
- Esso ebbe origine negli anni ’40, ma ha ottenuto un significativo studio solo in tempi relativamente recenti, quando la tecnologia ha consentito di stampare circuiti in due e tre dimensioni a prezzi ragionevoli

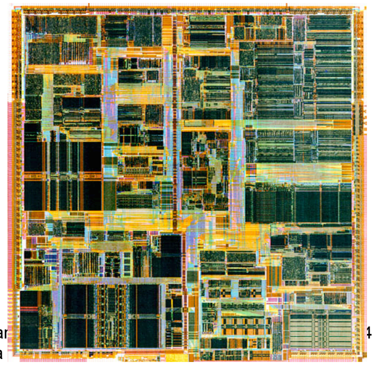
Il modello di Thompson (2)

Due esempi di circuiti stampati

Intel 2004



Intel Pentium



Il modello di Thompson (3)

La tecnologia di produzione VLSI impone molti vincoli al layout; in particolare, bisogna tenere conto:

- che la macchina che traccia i fili può solo approssimare segmenti obliqui con sequenze di segmenti orizzontali e verticali (⇒disegno ortogonale);
- che per evitare interferenze bisogna garantire che i fili siano ad una certa distanza l'uno dall'altro(⇒disegno su griglia);
- che i fili non si possono incrociare; per evitare gli incroci si possono far viaggiare i fili uno su una faccia e uno sull'altra, introducendo dei "buchi" per permettere il passaggio dei fili da una faccia all'altra; il numero di tali fori deve comunque essere contenuto poiché il costo di realizzazione è elevato (⇒minimizzazione degli incroci)
- …

Il modello di Thompson (3)

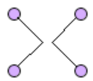
- … che i fili non devono essere troppo lunghi in quanto il ritardo di propagazione è proporzionale alla loro lunghezza e che, in caso di struttura a livelli, i fili dello stesso livello dovrebbero avere approssimativamente la stessa lunghezza, in modo da evitare problemi di sincronizzazione (⇒minimizzazione della lunghezza degli archi);
- che, di tutto il processo di produzione, il materiale è la cosa più costosa; per questo motivo è necessario che il layout abbia area minima(⇒minimizzazione dell'area).

N.B. similitudine con il disegno ortogonale su griglia

Il modello di Thompson (4)

Thompson ha introdotto un modello consistente con tutti questi vincoli:

- il layout di una topologia *G* è una rappresentazione su un insieme di tracce orizzontali e verticali a distanza unitaria l'una dall'altra che mappa:
  - i nodi del grafo nei punti di intersezione delle tracce,
  - gli archi del grafo in percorsi disgiunti costituiti da segmenti di tracce orizzontali e verticali; tali percorsi non possono intersecare nodi a loro non adiacenti e possono avere intersezioni tra loro solo in corrispondenza degli incroci;
  - non sono permesse sovrapposizioni, non sono permessi incroci arco-nodo, non sono permessi "knock-knees"



Il modello di Thompson (5)

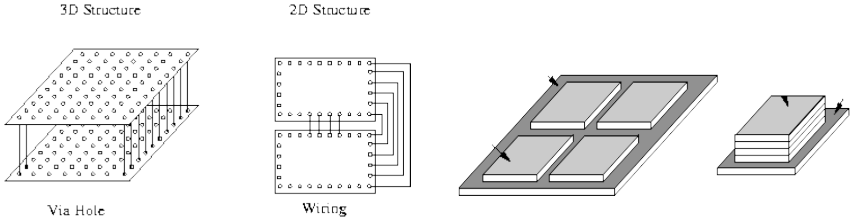
- Dunque, il layout è un disegno ortogonale su griglia per cui si richiede la minimizzazione dell'area, del numero di incroci e della lunghezza degli archi.
- Usiamo gli algoritmi già visti per disegnare in modo ortogonale su griglia un grafo?
- questa tecnica non funziona in modo soddisfacente: gli algoritmi per disegnare grafi garantiscono certe limitazioni delle funzioni di ottimizzazione per OGNI grafo dato in input che soddisfi i requisiti necessari; le topologie di interconnessione, però sono grafi con forte struttura (tipicamente regolari, molto simmetriche, con proprietà ricorsive, …) e, sfruttando queste proprietà, si possono ottenere risultati migliori.

Il modello di Thompson (6)

- Gli algoritmi per disegnare grafi prendono in input un grafo e lo disegnano
- Gli algoritmi per il layout sono progettati per una peculiare topologia di interconnessione e, perciò, ne prendono in input la dimensione.
- N.B. L'ottimizzazione al meglio delle funzioni di ottimizzazione (in special modo l'area), anche nelle costanti, è un obiettivo assai più importante che nel caso di disegni di grafi per quanto già osservato: se un layout viene fatto nella metà dell'area rispetto ad un altro, il suo costo sarà dimezzato!

Cenni al layout 3D

- Negli ultimi anni si è sempre più diffuso il layout 3D: la topologia è rappresentata su una serie di lastre sovrapposte
- ulteriore ottimizzazione della lunghezza dei fili e del # di svolte
- diminuzione dell'uso del silicio



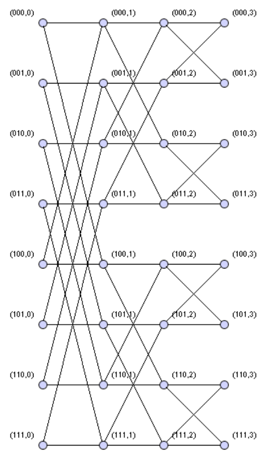
Layout della Butterfly

La Butterfly (1)

Def. Sia $N=2^n$ (e quindi $n=\log N$); una Butterfly *n*-dimensionale è un grafo livellato con $N\ (n+1)$ nodi (n+1 livelli ognuno con 2^n nodi) e $2Nn$ archi, i cui nodi sono etichettati con una coppia (i, w), dove i è il *livello* del nodo e w è un numero binario di n bit che indica la riga del nodo.

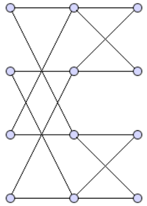
Due nodi (i,w) e (i', w') sono adiacenti se e solo se $i'=i+1$ e:

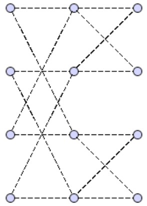
- $w=w'$ (*straight edge*) o
- w e w' differiscono esattamente nell'i-esimo bit (*cross edge*).



La Butterfly (2)

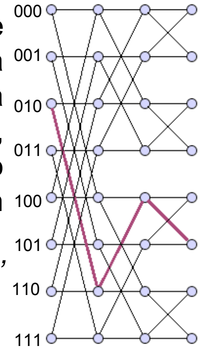
- I nodi della Butterfly sono *crossbar switches,* cioè switches che hanno due ingressi e due uscite e possono assumere due stati, cross e bar.
- Possiamo quindi vedere la butterfly come una switching network che connette 2*N* ($N=2^n$) unità di input a 2*N* unità di output tramite una rete di *logN+1* livelli di *N* nodi ciascuno.
- La butterfly ha una struttura ricorsiva : una butterfly *n*-dimensionale contiene come sottografi due butterfly (*n*-1)-dimensionali che si possono ottenere rimovendo i nodi di livello *n*.





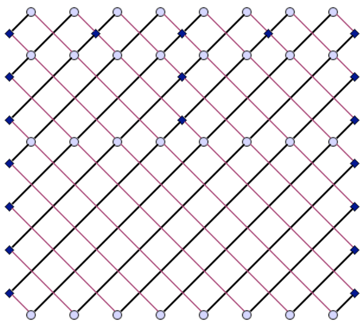
La Butterfly (3)

per ogni coppia di righe *w* e *w',* esiste uno ed un solo cammino di lunghezza *n* che collega un nodo di livello 0 sulla riga *w* e uno di livello *n* su una riga *w'*, questo cammino attraversa ogni livello esattamente una volta, usando un cross edge dal livello *i* al livello *i*+1 (*i*=0,…,*n*) se e solo se *w* e *w'* differiscono nell' i-esimo bit



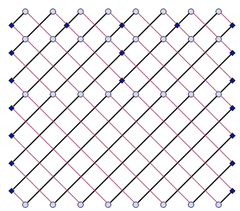
Layout di Wise (1)

Layout proposto da D.S.Wise ('81)



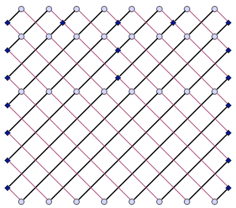
Layout di Wise (2)

- proprietà di avere tutti i fili di uno stesso livello della stessa lunghezza, dove la lunghezza cresce esponenzialmente con il livello
- la lunghezza di ogni cammino da un input ad un output è lineare in *N* (esattamente *2(N-1)*). Infatti:
  - consideriamo per semplicità il percorso dal nodo più in alto a sinistra a quello più in basso a destra
  - la lunghezza di questo cammino può essere calcolata come la diagonale di un quadrato di lato *√2 (N-1)*
  - la lunghezza di tale cammino è quindi uguale a *2(N-1)*.



Layout di Wise (3)

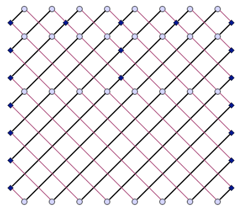
- Il layout è formato da due strati, stampati sulle due facce della lastra di silicio; su uno si trovano le linee che vanno da in alto a sinistra a in basso a destra (linee rosse), sull'altro si trovano i fili che vanno da in alto a sinistra a in basso a destra (linee nere), si può pensare a questi due strati come alle due facce della lastra di supporto; questo posizionamento dei fili in due strati è necessario per evitare gli incroci tra i fili.
- Alcuni fili formano dei "knock-knees", che non sono permessi; affinché la prossimità tra questi fili non causi interferenze è necessario inserire dei dispositivi speciali, rappresentati come dei piccoli quadrati.



Layout di Wise (4)

PROPRIETA':

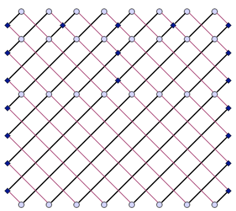
- Vantaggi:
  - Area buona $\sqrt{2}\,(N-1) \times \sqrt{2}\,(N-1) = 2N^2+o(N^2)$
  - lunghezza dei fili costante su ogni livello; questo non è vero in ogni layout: nel disegno classico della butterfly, ad esempio, sull'ultimo livello, gli straight-edges hanno lunghezza unitaria, mentre i cross-edges hanno lunghezza lineare in *N*; questa è una proprietà estremamente negativa perché si perde il sincronismo del flusso delle informazioni;
  - i nodi di input ed output giacciono sul contorno, che può essere richiesto da alcune applicazioni.



Layout di Wise (5)

PROPRIETA' (segue):

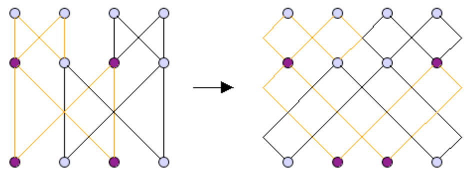
- Svantaggi:
  - linee "slanted", quindi l'area misurata non è quella del rettangolo con lati paralleli agli assi cartesiani, ma con i lati a 45°; seguendo la definizione standard di area essa risulta $2(N-1) \times 2(N-1) = 4N^2+o(N^2)$, infatti il quadrato circoscritto al layout con lati paralleli alle linee della griglia ha lato di lunghezza pari al cammino dal nodo più in alto a sinistra a quello più in basso a destra e cioè $2(N-1)$;
  - non è un layout 'pulito', poiché i "knock-knees" non vengono evitati ma sistemati con dei dispositivi che, oltre tutto, non hanno area nulla, quindi il loro inserimento causa un peggioramento delle dimensioni del layout;



Layout di Wise (6)

PROPRIETA' (segue):

- Svantaggi:
  - per ottenere il layout proposto bisogna permutare l'ordine dei nodi

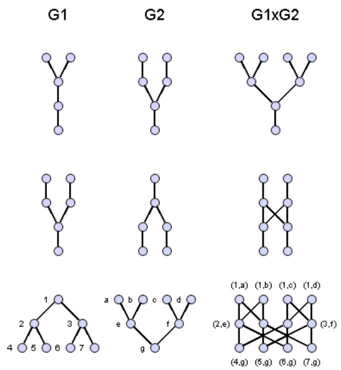


Layout di Even e Even (1)

- Layout presentato da Even e Even ('00), basato sulla nozione di Layered Cross Product
- **Def.** Un *grafo livellato con l+1 livelli* $G=(V_0, V_1, \ldots, V_l, E)$ consiste di l+1 livelli di nodi; V_i è l'insieme (non vuoto) di nodi di livello i; E è l'insieme degli archi: ogni arco (u,v) connette due nodi di livelli adiacenti, cioè, se u è al livello i allora v è al livello $i+1$.
- **Def**. (Evene & Litman '92) Il *Layered Cross Product* (*LCP*) di due grafi livellati di $l+1$ livelli, $G^1=(V_0^1, V_1^1, \ldots, V_l^1, E^1)$ e $G^2=(V_0^2, V_1^2, \ldots, V_l^2, E^2)$, è un grafo livellato con l+1 livelli $G=(V_0, V_1, \ldots, V_l, E)$ tale che:
  - per ogni $i=0, \ldots, l$ $V_1 = V_1^1 \times V_1^2$ (ogni livello è il prodotto cartesiano dei livelli dei grafi fattore);
  - esiste un arco (u,v) che connette i nodi (u^1,u^2) e (v^1,v^2) se e solo se (u^1,v^1) ed (u^2,v^2) sono archi dei grafi fattore.

Layout di Even e Even (2)

Esempi di LCP



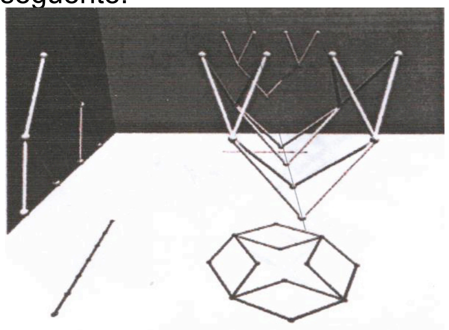
Even e Litman hanno dimostrato che molte topologie di interconnessione ben note sono il risultato del prodotto di semplici grafi livellati come gli alberi.

In particolare, la butterfly è il LCP di due alberi binari completi uno con la radice posta in alto, l'altro con la radice posta in basso.

Layout di Even e Even (3)

Metodo di proiezione:

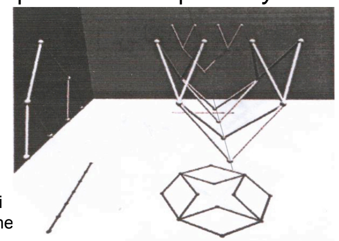
- Siano G^1 e G^2 due grafi livellati di $l+1$ livelli e sia G il loro LCP. Un disegno bidimensionale di G si ottiene nel modo seguente:



Layout di Even e Even (4)

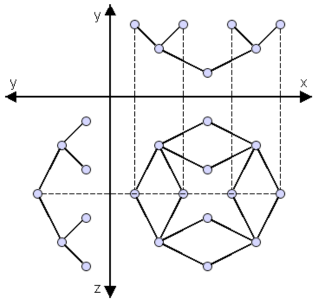
Metodo di proiezione (segue):

- Considera lo spazio cartesiano *xyz*;
- Disegna sul piano *xy* il grafo G^1 e sul piano *yz* il grafo G^2 in modo tale che la coordinata y di ogni nodo di livello i sia uguale a i, le altre coordinate siano interi;
- Costruisci un disegno 3D di G come segue: se $u \in V_i^1$ è disegnato alle coordinate $(x_u, i, 0)$ e $v \in V_i^2$ è disegnato alle coordinate $(0, i, z_v)$, allora le coordinate del nodo (u,v) sono (x_u, i, z_v), cioè i nodi di G sono i punti di intersezione delle rette perpendicolari al piano *xy* passanti per i nodi di G^1 e delle rette perpendicolari al piano *yz* passanti per i nodi di G^2.
- Il disegno 2D di G si ottiene proiettando il disegno 3D sul piano *xz*



Layout di Even e Even (5)

Oss. Si può evitare di passare per la rappresentazione 3D, infatti i nodi del LCP coincidono con le intersezioni dei prolungamenti sul piano *xz* delle proiezioni dei nodi di livello *i* di G^1 sull'asse *x* e dei nodi di livello *i* di G^2 sull'asse *z* per $i=0,\ldots,l$



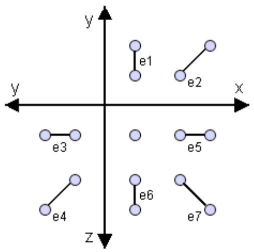
Layout di Even e Even (6)

- Il metodo di proiezione può produrre rappresentazioni che non rispettano i vincoli dettati dal layout di topologie, ad esempio il disegno costruito pur essendo su griglia non è ortogonale.
- Analizziamo i diversi tipi di archi risultanti dall'LCP; ciò ci permetterà di imporre le condizioni necessarie e sufficienti affinché il metodo di proiezione produca archi che corrono lungo le linee della griglia, nodi mappati in punti distinti della griglia e archi che non si sovrappongono.

Layout di Even e Even (7)

Ci sono quattro tipi di archi nel grafo prodotto *G*:

1. il prodotto di due archi obliqui dà come risultato un arco obliquo;
2. il prodotto di un arco verticale ed uno obliquo dà come risultato un arco verticale;
3. il prodotto di un arco obliquo ed uno verticale dà come risultato un arco orizzontale;
4. il prodotto di due archi verticali mappa due nodi distinti in uno stesso punto della griglia.



Layout di Even e Even (8)

- Per ottenere un layout ammissibile tramite il metodo di proiezione bisogna imporre che non ci siano archi generati dal prodotto di due archi obliqui o di due archi verticali. Più precisamente, possiamo formalizzare al modo seguente:
- ***1.*** *Il metodo di proiezione genera un layout di G i cui archi giacciono su linee della griglia se e solo se i disegni di* G^1 *e* G^2 *soddisfano la seguente condizione: per ciascun arco di G, esattamente uno dei suoi fattori è obliquo.*
- Questa affermazione previene anche le sovrapposizioni di nodi nell'ambito di ciascun livello.

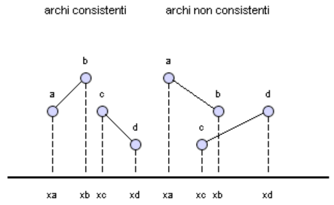
Layout di Even e Even (9)

- E’ necessario imporre anche che nodi di diversi livelli non si sovrappongono. Per fare questo, è sufficiente che sia verificata la seguente affermazione:

- ***2.*** *Il metodo di proiezione genera un layout di G in cui al più un nodo è mappato su ogni punto della griglia se e solo se, per ogni $i=0, 1, \ldots, l$, gli insiemi $\{(x_u, z_v): u \in V_i^1 \text{ e } v \in V_i^2\}$ sono disgiunti.*

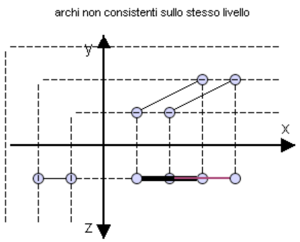
Layout di Even e Even (10)

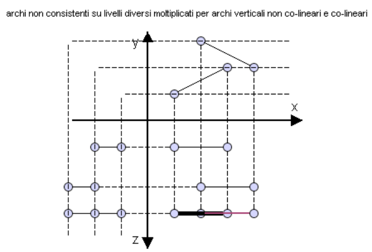
- Si considerino ora due archi obliqui *(a,b)* e *(c,d)* in G^1; i quattro nodi hanno coordinate:
  - nodo *a*: $(x_a, i, 0)$;
  - nodo *b*: $(x_b, i+1, 0)$;
  - nodo *c*: $(x_c, j, 0)$;
  - nodo *d*: $(x_d, j+1, 0)$.
- I due archi si dicono *consistenti* se e solo se gli intervalli aperti (x_a, x_b) e (x_c, x_d) sono disgiunti.



Layout di Even e Even (11)

- ***3.*** *Il metodo di proiezione genera un layout di G in cui nessun arco di griglia viene usato due volte se e solo se per ogni coppia di archi inconsistenti in G^1 (G^2):*
  - *i due archi non sono sullo stesso livello e*
  - *sui due livelli su cui si trovano non esistono due archi co-lineari in G^2 (G^1)*

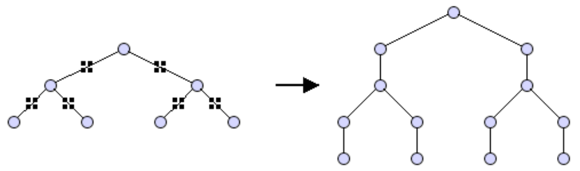




Layout di Even e Even (12)

- Affinché il metodo di proiezione produca un layout ammissibile del grafo *G* dobbiamo quindi disegnare i due grafi fattori in modo che siano rispettate le tre affermazioni enunciate.
- ***1.*** *Il metodo di proiezione genera un layout di G i cui archi giacciono su linee della griglia se e solo se i disegni di G^1 e G^2 soddisfano la seguente condizione: per ciascun arco di G, esattamente uno dei suoi fattori è obliquo.*
- Introduciamo nel disegno dei due grafi fattori dei nodi e degli archi fittizi in modo tale che:
  - ogni arco sia spezzato in due archi uno obliquo e uno verticale;
  - nel disegno di G^1 gli archi obliqui siano nei livelli dispari e quelli verticali nei livelli pari;
  - nel disegno di G^2 gli archi obliqui siano nei livelli pari e quelli verticali nei livelli dispari.

Layout di Even e Even (13)

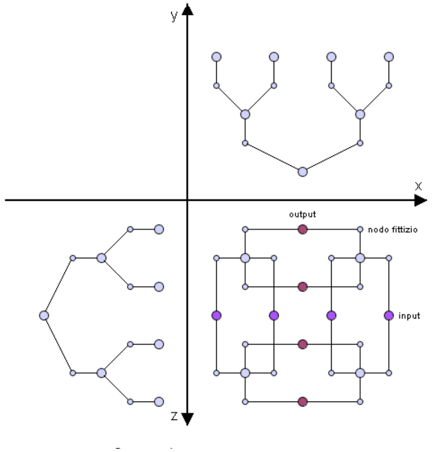


In questo modo viene simulata la creazione delle svolte.

Layout di Even e Even (14)

- ***2.** Il metodo di proiezione genera un layout di G in cui al più un nodo è mappato su ogni punto della griglia se e solo se, per ogni i=0, 1, …, l, gli insiemi $\{(x_u, z_v): u \in V_i^1$ e $v \in V_i^2\}$ sono disgiunti.*
- Bisogna fare in modo che non ci siano due nodi, esclusi gli estremi di un arco verticale, nel disegno di ciascun fattore, che condividano la stessa coordinata; questo è sempre possibile allargando opportunamente il disegno.
- ***3.** Il metodo di proiezione genera un layout di G in cui nessun arco di griglia viene usato due volte se e solo se per ogni coppia di archi inconsistenti in G^1 (G^2):*
  - *i due archi non sono sullo stesso livello e*
  - *sui due livelli su cui si trovano non esistono due archi co-lineari in G^2 (G^1)*
- Molto difficile in caso di grafi qualsiasi (collo di bottiglia di questo metodo). Nel caso della butterfly comunque, questo può essere fatto.

Layout di Even e Even (15)

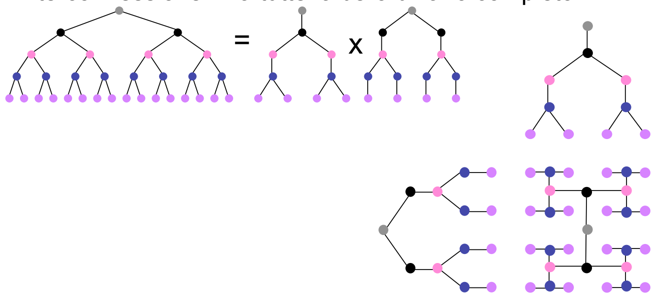


Layout di Even e Even (16)

- Otteniamo il layout della *n*-butterfly come prodotto di due alberi binari completi di *n*+1 livelli, uno downward e l'altro upward come segue:
  - nel disegno dell'albero binario, aggiungere i nodi fittizi ed assegnare ad ogni nodo una colonna per evitare collisioni di nodi;
  - disegnare un albero sul piano *xy*, l'altro sul piano *yz*;
  - (opzionale) costruire il LCP in 3D;
  - proiettare il LCP sul piano *xz*.
- Il layout così ottenuto ha le seguenti proprietà:
  - è simmetrico;
  - ha altezza $H=2(N-1)$;
  - ha ampiezza $W=2(N-1)$;
  - ha area $4N^2+o(N^2)$;
  - i nodi di input e output non sono sui bordi (proprietà negativa…)
  - tutti gli archi di ciascun livello hanno la stessa lunghezza

Layout di Even e Even (17)

- Con questa tecnica si possono rappresentare parecchie topologie di interconnessione. Tra tutte: albero binario completo:



Raffronto delle tecniche

- WISE - PRO:
  - area relativamente bassa
  - “sembrare” una butterfly
  - nodi di input/ output sul bordo
- WISE - CONTRO:
  - knok-knees
  - griglia “slanted”
- EVEN & EVEN - PRO:
  - elimina i difetti
- EVEN & EVEN - CONTRO:
  - aumenta l’area
  - pone i nodi di input/output all’interno del disegno

Altri risultati

- Nell’ottica di ottimizzare al meglio l’area, sono stati proposti altri latyouts della butterfly:
  - Dinitz (‘98) dimostra che il layout di Even ed Even può essere migliorato, con degli aggiustamenti locali in modo da avere un’area pari a $11/6\ N^2+o(N^2)$
  - Successivamente, Avior et al. (‘98) dimostrano che un layout con rettangolo circoscritto a lati paralleli agli assi cartesiani non può avere area minore di $N^2 + o(N^2)$ e forniscono un algoritmo che produce un layout di area ottima
  - Dinitz et al. (‘99) dimostrano infine che, se si accetta che il layout possa avere rettangolo circoscritto con i lati non necessariamente paralleli agli assi (cioè ‘slanted’), allora area $1/2\ N^2+o(N^2)$ è necessaria e sufficiente
- Questi lavori chiudono la problematica del layout di area ottima della butterfly.