

UNIVERSITA' DEGLI STUDI DI PISA

Facoltà di Ingegneria

Corso di Laurea in Ingegneria Informatica

Tesi di Laurea

**PROGETTAZIONE LOGICA E DEFINIZIONE
DI PROTOCOLLI E POLITICHE PER LE
CACHES D-NUCA TRIANGOLARI**

Relatori:

Ch.mo Prof. Cosimo Antonio Prete

Ch.mo Prof. Gianluca Dini

Candidato:

Simone Grechi

Anno Accademico 2002/2003

ABSTRACT

Gli odierni processori per sistemi ad alte prestazioni incorporano caches di dimensioni sempre crescenti. Ad esempio l'Alpha 21364 ha 1.75MB di L2, l'HP PA-8700 contiene 2.25MB di cache unificata e l'Itanium2 prevede 3MB di cache L3. Dato che la cache L2 occupa circa la metà dell'area di silicio del chip (il 30% nell'Alpha 21264, il 60% nello StrongARM), è facile capire come questo componente incida non poco sul costo dell'intero processore e sulle sue prestazioni.

In questo lavoro di tesi saranno valutati differenti progetti di architetture di cache con l'obiettivo di migliorarne le prestazioni in termini di tempi di accesso, capacità, costo e risparmio energetico. In particolare sarà presentata una architettura di memoria cache con tempo di accesso che è funzione degli indirizzi dei dati e su questa saranno costruite alcune varianti sempre con riguardo alle relative prestazioni. Inoltre per ogni configurazione presentata saranno fatte varie prove di test in modo da validare il progetto e renderlo credibile.

SOMMARIO

1	INTRODUZIONE.....	1
2	STATO DELL'ARTE	4
2.1	<i>UCA CACHES.....</i>	5
2.2	<i>NUCA CACHES.....</i>	8
2.2.1	<i>Private Channels S-NUCA.....</i>	9
2.2.2	<i>Switched Channels S-NUCA.....</i>	12
2.3	<i>D-NUCA CACHES.....</i>	14
2.3.1	<i>Mappaggio dei blocchi</i>	14
2.3.2	<i>Ricerca dei dati.....</i>	16
2.3.3	<i>Migrazione dei blocchi</i>	17
3	CACHES D-NUCA TRIANGOLARI.....	24
3.1	<i>DEFINIZIONE.....</i>	25
3.2	<i>MAPPAGGIO DEI BLOCCHI</i>	30
3.3	<i>MAPPAGGIO VERTICALE DEI BLOCCHI.....</i>	33
3.4	<i>ROUTING.....</i>	37
3.5	<i>POLITICHE DI RIMPIAZZAMENTO E PROMOZIONE.....</i>	41
3.6	<i>TECNICHE DI RICERCA</i>	43
3.7	<i>TECNICA PER LA RIDUZIONE DELLE PERDITE DELL'ENERGIA STATICA</i>	44
3.8	<i>TD-NUCA DECRESCENTI.....</i>	45
4	GLI STRUMENTI USATI.....	47
4.1	<i>IL SIMULATORE: D-NUCA SIM-ALPHA 1.0.....</i>	47
4.2	<i>I BENCHMARKS: SPECINT 2000.....</i>	49
4.3	<i>CONFIGURAZIONE DI BASE DELL'ARCHITETTURA.....</i>	51
4.4	<i>IL SIMULATORE MODIFICATO: TD-NUCA SIM-ALPHA</i>	54
5	VALUTAZIONE DELLE PRESTAZIONI	56
5.1	<i>VALIDAZIONE.....</i>	56
5.1.1	<i>Modalità non-interlacciata ed a conflitti disabilitati.....</i>	58
5.1.2	<i>Modalità interlacciata.....</i>	63
5.2	<i>CONFIGURAZIONE DI BASE.....</i>	68
5.3	<i>CONFIGURAZIONE AVANZATA.....</i>	78

6	CONCLUSIONI.....	82
A	PRESTAZIONI DEI SINGOLI BENCHMARKS	84
<i>A.1</i>	<i>176.gcc.....</i>	<i>84</i>
<i>A.2</i>	<i>181.mcf.....</i>	<i>94</i>
<i>A.3</i>	<i>256.bzip2.....</i>	<i>103</i>
<i>A.4</i>	<i>300.twolf.....</i>	<i>112</i>

INDICE DELLE FIGURE

Figura 1: <i>organizzazione interna di una cache Private Channels S-NUCA.</i>	9
Figura 2: <i>organizzazione interna di una cache Switched Channels S-NUCA.</i>	12
Figura 3: <i>mappaggio dei bank sets nei banchi.</i>	15
Figura 4: <i>struttura dei banchi di una cache D-NUCA Triangolare a confronto con una D-NUCA rettangolare.</i>	26
Figura 5: <i>posizionamento di uno stesso blocco nelle 4 vie di una cache D-NUCA 8x4, e nelle 4 vie di una cache TD-NUCA [1,2,4,8].</i>	27
Figura 6: <i>suddivisione dell'indirizzo di un byte per una cache TD-NUCA.</i>	29
Figura 7: <i>possibili mapping dei blocchi nei banchi.</i>	30
Figura 8: <i>suddivisione dell'indirizzo e verso di lettura del campo bank nei due tipi di mapping.</i>	32
Figura 9: <i>modalità di distribuzione degli indirizzi in una via composta da due banchi.</i>	35
Figura 10: <i>combinazioni tra le modalità di mapping orizzontale con quello verticale. Per ognuna di esse è mostrata la distribuzione degli indirizzi nei banchi, e la funzione usata per effettuare la scelta del banco.</i>	36
Figura 11: <i>diversi routing nelle due caches.</i>	38
Figura 12: <i>esempio di accesso ad un bank set (in rosso) nei due diversi routing.</i>	40
Figura 13: <i>simple mapping e fair mapping per una TD-NUCA decrescente.</i>	46

INDICE DELLE TABELLE

Tabella 1: <i>IPC e miss rate di una cache UCA al variare della sua dimensione.</i>	6
Tabella 2: <i>IPC, unloaded latency e loaded latency di una Private Channels S-NUCA al variare della sua dimensione.</i>	10
Tabella 3: <i>IPC e latenze di una Switched Channels S-NUCA al variare della sua dimensione.</i>	13
Tabella 4: <i>latenza media, IPC e miss rate di una D-NUCA al variare della sua dimensione.</i>	20
Tabella 5: <i>valutazione delle singole politiche avanzate della D-NUCA.</i>	22
Tabella 6: <i>FFWD e RUN per i 4 SpecInt 2000 utilizzati.</i>	49
Tabella 7: <i>principali parametri di configurazione dell'architettura Alpha 21264 usata nei tests di prestazione del capitolo 5.</i>	51
Tabella 8: <i>comparazione di una cache rettangolare da 16MB in modalità non-interlacciata e con la disabilitazione dei conflitti di canale nelle due configurazioni: D-NUCA 16x16 (test n. 1) e TD-NUCA [16,16,16,16,16,16,16,16,16,16,16,16,16,16] (test n. 2).</i>	59
Tabella 9: <i>comparazione di una cache rettangolare da 8MB in modalità non-interlacciata e con la disabilitazione dei conflitti di canale nelle due configurazioni: D-NUCA 16x8 (test n. 3) e TD-NUCA [16,16,16,16,16,16,16,16] (test n. 4).</i>	60
Tabella 10: <i>comparazione di una cache rettangolare da 4MB in modalità non-interlacciata e con la disabilitazione dei conflitti di canale nelle due configurazioni: D-NUCA 8x4 (test n. 5) e TD-NUCA [8,8,8,8] (test n. 6).</i>	61
Tabella 11: <i>comparazione di una cache rettangolare da 2MB in modalità non-interlacciata e con la disabilitazione dei conflitti di canale nelle due configurazioni: D-NUCA 4x4 (test n. 7) e TD-NUCA [4,4,4,4] (test n. 8).</i>	61
Tabella 12: <i>comparazione di una cache rettangolare da 16MB in modalità interlacciata nelle due configurazioni: D-NUCA 16x16 (test n. 11) e TD-NUCA [16,16,16,16,16,16,16,16,16,16,16,16,16,16] (test n. 12).</i>	64
Tabella 13: <i>comparazione di una cache rettangolare da 8MB in modalità interlacciata nelle due configurazioni: D-NUCA 16x8 (test n. 13) e TD-NUCA [16,16,16,16,16,16,16,16] (test n. 14).</i>	65

Tabella 14: <i>comparazione di una cache rettangolare da 4MB in modalità interlacciata nelle due configurazioni: D-NUCA 8x4 (test n. 15) e TD-NUCA [8,8,8,8] (test n. 16).</i>	66
Tabella 15: <i>comparazione di una cache rettangolare da 2MB in modalità interlacciata nelle due configurazioni: D-NUCA 4x4 (test n. 17) e TD-NUCA [4,4,4,4] (test n. 18).</i>	66
Tabella 16: <i>comparazione di una cache rettangolare da 16MB con le equivalenti triangolari crescente e decrescente da circa 8MB.</i>	69
Tabella 17: <i>comparazione di una cache rettangolare da 8MB con le equivalenti triangolari crescente e decrescente da circa 4MB.</i>	70
Tabella 18: <i>comparazione di una cache rettangolare da 4MB con le equivalenti triangolari crescente e decrescente da circa 2MB.</i>	71
Tabella 19: <i>comparazione di una cache rettangolare da 2MB con le equivalenti triangolari crescente e decrescente da circa 1MB.</i>	73
Tabella 20: <i>comparazione di una cache rettangolare da 1MB con le equivalenti triangolari crescente e decrescente da circa 512KB.</i>	74
Tabella 21: <i>comparazione di una cache rettangolare da 1MB con una cache triangolare decrescente da 1MB.</i>	75
Tabella 22: <i>comparazione della configurazione base di caches triangolari da 1MB con alcune configurazioni avanzate della stessa.</i>	78
Tabella 23: <i>comparazione della configurazione base di caches triangolari da 1MB con alcune configurazioni avanzate della stessa (segue).</i>	79
Tabella 24: <i>comparazione di una cache rettangolare da 16MB in modalità non-interlacciata e con la disabilitazione dei conflitti di canale nelle due configurazioni: D-NUCA 16x16 (test n. 1) e TD-NUCA [16,16,16,16,16,16,16,16,16,16,16,16,16,16,16,16] (test n. 2).</i>	84
Tabella 25: <i>comparazione di una cache rettangolare da 8MB in modalità non-interlacciata e con la disabilitazione dei conflitti di canale nelle due configurazioni: D-NUCA 16x8 (test n. 3) e TD-NUCA [16,16,16,16,16,16,16,16] (test n. 4).</i>	85
Tabella 26: <i>comparazione di una cache rettangolare da 4MB in modalità non-interlacciata e con la disabilitazione dei conflitti di canale nelle due configurazioni: D-NUCA 8x4 (test n. 5) e TD-NUCA [8,8,8,8] (test n. 6).</i>	85

Tabella 27: <i>comparazione di una cache rettangolare da 2MB in modalità non-interlacciata e con la disabilitazione dei conflitti di canale nelle due configurazioni: D-NUCA 4x4 (test n. 7) e TD-NUCA [4,4,4,4] (test n. 8).</i>	86
Tabella 28: <i>comparazione di una cache rettangolare da 16MB in modalità interlacciata nelle due configurazioni: D-NUCA 16x16 (test n. 11) e TD-NUCA [16,16,16,16,16,16,16,16,16,16,16,16,16,16] (test n. 12).</i>	86
Tabella 29: <i>comparazione di una cache rettangolare da 8MB in modalità interlacciata nelle due configurazioni: D-NUCA 16x8 (test n. 13) e TD-NUCA [16,16,16,16,16,16,16,16] (test n. 14).</i>	87
Tabella 30: <i>comparazione di una cache rettangolare da 4MB in modalità interlacciata nelle due configurazioni: D-NUCA 8x4 (test n. 15) e TD-NUCA [8,8,8,8] (test n. 16).</i>	87
Tabella 31: <i>comparazione di una cache rettangolare da 2MB in modalità interlacciata nelle due configurazioni: D-NUCA 4x4 (test n. 17) e TD-NUCA [4,4,4,4] (test n. 18).</i>	88
Tabella 32: <i>comparazione di una cache rettangolare da 16MB con le equivalenti triangolari crescente e decrescente da circa 8MB.</i>	88
Tabella 33: <i>comparazione di una cache rettangolare da 8MB con le equivalenti triangolari crescente e decrescente da circa 4MB.</i>	89
Tabella 34: <i>comparazione di una cache rettangolare da 4MB con le equivalenti triangolari crescente e decrescente da circa 2MB.</i>	89
Tabella 35: <i>comparazione di una cache rettangolare da 2MB con le equivalenti triangolari crescente e decrescente da circa 1MB.</i>	90
Tabella 36: <i>comparazione di una cache rettangolare da 1MB con le equivalenti triangolari crescente e decrescente da circa 512KB.</i>	90
Tabella 37: <i>comparazione della configurazione base di caches triangolari da 2MB con alcune configurazioni avanzate della stessa.</i>	92
Tabella 38: <i>comparazione della configurazione base di caches triangolari da 2MB con alcune configurazioni avanzate della stessa (segue).</i>	92

Tabella 39: <i>comparazione di una cache rettangolare da 16MB in modalità non-interlacciata e con la disabilitazione dei conflitti di canale nelle due configurazioni: D-NUCA 16x16 (test n. 1) e TD-NUCA [16,16,16,16,16,16,16,16,16,16,16,16,16,16,16,16] (test n. 2)</i>	94
Tabella 40: <i>comparazione di una cache rettangolare da 8MB in modalità non-interlacciata e con la disabilitazione dei conflitti di canale nelle due configurazioni: D-NUCA 16x8 (test n. 3) e TD-NUCA [16,16,16,16,16,16,16,16] (test n. 4)</i>	94
Tabella 41: <i>comparazione di una cache rettangolare da 4MB in modalità non-interlacciata e con la disabilitazione dei conflitti di canale nelle due configurazioni: D-NUCA 8x4 (test n. 5) e TD-NUCA [8,8,8,8] (test n. 6)</i>	95
Tabella 42: <i>comparazione di una cache rettangolare da 2MB in modalità non-interlacciata e con la disabilitazione dei conflitti di canale nelle due configurazioni: D-NUCA 4x4 (test n. 7) e TD-NUCA [4,4,4,4] (test n. 8)</i>	95
Tabella 43: <i>comparazione di una cache rettangolare da 16MB in modalità interlacciata nelle due configurazioni: D-NUCA 16x16 (test n. 11) e TD-NUCA [16,16,16,16,16,16,16,16,16,16,16,16,16,16,16,16] (test n. 12)</i>	96
Tabella 44: <i>comparazione di una cache rettangolare da 8MB in modalità interlacciata nelle due configurazioni: D-NUCA 16x8 (test n. 13) e TD-NUCA [16,16,16,16,16,16,16,16] (test n. 14)</i>	96
Tabella 45: <i>comparazione di una cache rettangolare da 4MB in modalità interlacciata nelle due configurazioni: D-NUCA 8x4 (test n. 15) e TD-NUCA [8,8,8,8] (test n. 16)</i>	97
Tabella 46: <i>comparazione di una cache rettangolare da 2MB in modalità interlacciata nelle due configurazioni: D-NUCA 4x4 (test n. 17) e TD-NUCA [4,4,4,4] (test n. 18)</i>	97
Tabella 47: <i>comparazione di una cache rettangolare da 16MB con le equivalenti triangolari crescente e decrescente da circa 8MB</i>	98
Tabella 48: <i>comparazione di una cache rettangolare da 8MB con le equivalenti triangolari crescente e decrescente da circa 4MB</i>	98
Tabella 49: <i>comparazione di una cache rettangolare da 4MB con le equivalenti triangolari crescente e decrescente da circa 2MB</i>	99
Tabella 50: <i>comparazione di una cache rettangolare da 2MB con le equivalenti triangolari crescente e decrescente da circa 1MB</i>	99

Tabella 51: <i>comparazione di una cache rettangolare da 1MB con le equivalenti triangolari crescente e decrescente da circa 512KB.</i>	100
Tabella 52: <i>comparazione della configurazione base di caches triangolari da 2MB con alcune configurazioni avanzate della stessa.</i>	101
Tabella 53: <i>comparazione della configurazione base di caches triangolari da 2MB con alcune configurazioni avanzate della stessa (segue).</i>	101
Tabella 54: <i>comparazione di una cache rettangolare da 16MB in modalità non-interlacciata e con la disabilitazione dei conflitti di canale nelle due configurazioni: D-NUCA 16x16 (test n. 1) e TD-NUCA [16,16,16,16,16,16,16,16,16,16,16,16,16,16] (test n. 2).</i>	103
Tabella 55: <i>comparazione di una cache rettangolare da 8MB in modalità non-interlacciata e con la disabilitazione dei conflitti di canale nelle due configurazioni: D-NUCA 16x8 (test n. 3) e TD-NUCA [16,16,16,16,16,16,16,16] (test n. 4).</i>	103
Tabella 56: <i>comparazione di una cache rettangolare da 4MB in modalità non-interlacciata e con la disabilitazione dei conflitti di canale nelle due configurazioni: D-NUCA 8x4 (test n. 5) e TD-NUCA [8,8,8,8] (test n. 6).</i>	104
Tabella 57: <i>comparazione di una cache rettangolare da 2MB in modalità non-interlacciata e con la disabilitazione dei conflitti di canale nelle due configurazioni: D-NUCA 4x4 (test n. 7) e TD-NUCA [4,4,4,4] (test n. 8).</i>	104
Tabella 58: <i>comparazione di una cache rettangolare da 16MB in modalità interlacciata nelle due configurazioni: D-NUCA 16x16 (test n. 11) e TD-NUCA [16,16,16,16,16,16,16,16,16,16,16,16,16,16] (test n. 12).</i>	105
Tabella 59: <i>comparazione di una cache rettangolare da 8MB in modalità interlacciata nelle due configurazioni: D-NUCA 16x8 (test n. 13) e TD-NUCA [16,16,16,16,16,16,16,16] (test n. 14).</i>	105
Tabella 60: <i>comparazione di una cache rettangolare da 4MB in modalità interlacciata nelle due configurazioni: D-NUCA 8x4 (test n. 15) e TD-NUCA [8,8,8,8] (test n. 16).</i>	106
Tabella 61: <i>comparazione di una cache rettangolare da 2MB in modalità interlacciata nelle due configurazioni: D-NUCA 4x4 (test n. 17) e TD-NUCA [4,4,4,4] (test n. 18).</i>	106
Tabella 62: <i>comparazione di una cache rettangolare da 16MB con le equivalenti triangolari crescente e decrescente da circa 8MB.</i>	107

Tabella 63: <i>comparazione di una cache rettangolare da 8MB con le equivalenti triangolari crescente e decrescente da circa 4MB.</i>	107
Tabella 64: <i>comparazione di una cache rettangolare da 4MB con le equivalenti triangolari crescente e decrescente da circa 2MB.</i>	108
Tabella 65: <i>comparazione di una cache rettangolare da 2MB con le equivalenti triangolari crescente e decrescente da circa 1MB.</i>	108
Tabella 66: <i>comparazione di una cache rettangolare da 1MB con le equivalenti triangolari crescente e decrescente da circa 512KB.</i>	109
Tabella 67: <i>comparazione della configurazione base di caches triangolari da 2MB con alcune configurazioni avanzate della stessa.</i>	110
Tabella 68: <i>comparazione della configurazione base di caches triangolari da 2MB con alcune configurazioni avanzate della stessa (segue).</i>	110
Tabella 69: <i>comparazione di una cache rettangolare da 16MB in modalità non-interlacciata e con la disabilitazione dei conflitti di canale nelle due configurazioni: D-NUCA 16x16 (test n. 1) e TD-NUCA [16,16,16,16,16,16,16,16,16,16,16,16,16,16,16,16] (test n. 2).....</i>	112
Tabella 70: <i>comparazione di una cache rettangolare da 8MB in modalità non-interlacciata e con la disabilitazione dei conflitti di canale nelle due configurazioni: D-NUCA 16x8 (test n. 3) e TD-NUCA [16,16,16,16,16,16,16,16] (test n. 4).</i>	112
Tabella 71: <i>comparazione di una cache rettangolare da 4MB in modalità non-interlacciata e con la disabilitazione dei conflitti di canale nelle due configurazioni: D-NUCA 8x4 (test n. 5) e TD-NUCA [8,8,8,8] (test n. 6).....</i>	113
Tabella 72: <i>comparazione di una cache rettangolare da 2MB in modalità non-interlacciata e con la disabilitazione dei conflitti di canale nelle due configurazioni: D-NUCA 4x4 (test n. 7) e TD-NUCA [4,4,4,4] (test n. 8).....</i>	113
Tabella 73: <i>comparazione di una cache rettangolare da 16MB in modalità interlacciata nelle due configurazioni: D-NUCA 16x16 (test n. 11) e TD-NUCA [16,16,16,16,16,16,16,16,16,16,16,16,16,16,16,16] (test n. 12).....</i>	114
Tabella 74: <i>comparazione di una cache rettangolare da 8MB in modalità interlacciata nelle due configurazioni: D-NUCA 16x8 (test n. 13) e TD-NUCA [16,16,16,16,16,16,16,16] (test n. 14).</i>	114

Tabella 75: <i>comparazione di una cache rettangolare da 4MB in modalità interlacciata nelle due configurazioni: D-NUCA 8x4 (test n. 15) e TD-NUCA [8,8,8,8] (test n. 16).</i>	115
Tabella 76: <i>comparazione di una cache rettangolare da 2MB in modalità interlacciata nelle due configurazioni: D-NUCA 4x4 (test n. 17) e TD-NUCA [4,4,4,4] (test n. 18).</i>	115
Tabella 77: <i>comparazione di una cache rettangolare da 16MB con le equivalenti triangolari crescente e decrescente da circa 8MB.</i>	116
Tabella 78: <i>comparazione di una cache rettangolare da 8MB con le equivalenti triangolari crescente e decrescente da circa 4MB.</i>	116
Tabella 79: <i>comparazione di una cache rettangolare da 4MB con le equivalenti triangolari crescente e decrescente da circa 2MB.</i>	117
Tabella 80: <i>comparazione di una cache rettangolare da 2MB con le equivalenti triangolari crescente e decrescente da circa 1MB.</i>	117
Tabella 81: <i>comparazione di una cache rettangolare da 1MB con le equivalenti triangolari crescente e decrescente da circa 512KB.</i>	118
Tabella 82: <i>comparazione della configurazione base di caches triangolari da 2MB con alcune configurazioni avanzate della stessa.</i>	119
Tabella 83: <i>comparazione della configurazione base di caches triangolari da 2MB con alcune configurazioni avanzate della stessa (segue).</i>	119

INDICE DEI GRAFICI

Grafico 1: <i>IPC a confronto.</i>	76
Grafico 2: <i>accessi ai banchi a confronto.</i>	77
Grafico 3: <i>comparazione degli IPC fra le configurazioni di base ed avanzate (1MB).</i>	81
Grafico 4: <i>IPC a confronto per 176.gcc al variare della dimensione delle caches.</i>	91
Grafico 5: <i>comparazione degli IPC fra le configurazioni di base ed avanzate per 176.gcc (1MB).</i>	93
Grafico 6: <i>IPC a confronto per 181.mcf al variare della dimensione delle caches.</i>	100
Grafico 7: <i>comparazione degli IPC fra le configurazioni di base ed avanzate per 181.mcf (1MB).</i>	102
Grafico 8: <i>IPC a confronto per 256.bzip2 al variare della dimensione delle caches.</i>	109
Grafico 9: <i>comparazione degli IPC fra le configurazioni di base ed avanzate per 256.bzip2 (1MB).</i>	111
Grafico 10: <i>IPC a confronto per 300.twolf al variare della dimensione delle caches.</i>	118
Grafico 11: <i>comparazione degli IPC fra le configurazioni di base ed avanzate per 300.twolf (1MB).</i>	120

*Capitolo 1***INTRODUZIONE**

Con la corsa all'aumento della frequenza operativa dei processori attuali, i ritardi di propagazione dei segnali sono sempre più incidenti sulle latenze di ogni componente dei chips. La tecnologia attuale infatti riesce a minimizzare i tempi di risposta di ogni circuito ma, per limiti fisici, il ritardo introdotto dai canali di collegamento di ogni componente non può essere diminuito. Per capire meglio il problema del wire-delay pensiamo al tempo che impiega la luce per attraversare l'intero processore: con riferimento al pentium 4 [11] si ha che la lunghezza del core è 13,97mm, e data una velocità della luce nel silicio di circa 88.000.000m/s, il tempo di attraversamento risulta allora di 0,15875ns. La frequenza limite per far sì che un segnale non impieghi più di un ciclo di clock per l'attraversamento del circuito vale perciò 6,30GHz. Questo vuol dire che con frequenze operative superiori ai 6,30GHz un segnale impiegherà più di un ciclo di clock solo per attraversare l'intero processore. Il problema del wire-delay è affrontato nei moderni processori con le architetture a pipeline, le quali suddividono il chip in più segmenti ognuno dei quali opera in cicli di clock separati.

Il lavoro [4] spiega come è possibile ottimizzare il comportamento di un componente importante del processore con riguardo alla tecnica pipeline. Questo componente, il register file, è considerato come ultimo livello della gerarchia della memoria e serve a contenere le informazioni immediatamente utili al processore. Il suddetto lavoro prevede la strutturazione del register file in più livelli eterogenei di banchi. Questo componente infatti influisce fortemente sull'IPC (Instructions Per Cycle) ma, come è spiegato in [4], è possibile una sua suddivisione in più livelli ognuno

dei quali eterogeneo nel numero di registri, nel numero delle porte, e quindi nel tempo di accesso, e permettermi un accesso a pipeline. Ogni livello è poi acceduto con una politica LRU in modo da porre le informazioni più utili nei livelli veloci creando una sorta di cache nei primi livelli per le informazioni più usate. Si può evincere dagli studi fatti che l'architettura multiple-banked register file degrada le prestazioni del 10% (valori misurati con gli SpecInt 95) rispetto ad una architettura a singolo livello, ma che, fattorizzando i tempi di accesso a questo componente (in pratica accedendovi in modo pipeline) l'IPC viene innalzato dell'87%.

Come abbiamo appena visto, suddividere i circuiti in più livelli, cioè usare la tecnica a pipeline, permette di aumentare la frequenza di lavoro. Ma cosa succede per le caches interne al processore? Anche questi dispositivi soffrono il problema del wire-delay. Oggi si ipotizza un ritardo costante per l'accesso alla cache, senza considerare che in una cache a più vie, quelle più lontane dal controller avranno in futuro dei ritardi di propagazione più elevati e rallenteranno i tempi dell'intera cache. Quindi, alla luce delle considerazioni sopra esposte, un modello di cache che consideri tempi di accesso uniformi non sarà più realistico e già da oggi ci si sta muovendo in questo senso per cercare di limitare i tempi di accesso alle caches, ed in considerazione di ciò è stata riadattata a questo scopo una soluzione già esistente. L'idea di base sta nel creare un modello di cache nella quale i tempi di accesso sono non-uniformi e funzione del particolare banco di cache a cui si accede. Si viene così a creare un insieme di tempi di accesso ognuno dei quali è caratteristico di un particolare banco della cache. In pratica, le caches dovranno per forza di cose essere NUCA ovvero Non-Uniform Cache Access, in cui cioè ogni via ha un ritardo di accesso proprio ed indipendente dalle vie successive.

Come estensione a questi concetti è inoltre possibile pensare di ritardare volutamente i tempi di risposta delle vie finali della cache in modo da consentire un risparmio energetico. In considerazione dell'articolo [5] sulle

possibili tecnologie implementative dei circuiti elettronici, si può infatti costruire ogni via con una diversa tecnologia così da creare vie più veloci, ma che consumano di più, e vie più lente ma più risparmiuose.

Uno studio dell'Università del Texas ([2] e [6]) propone di creare un dinamismo fra le vie della cache in modo che i blocchi di dati in esse contenuti vengano gestiti con la politica LRU. Nel presente studio sarà ripreso questo concetto di Dynamic-NUCA e su questo tipo di caches valuteremo l'impatto della triangolarizzazione della loro geometria. In pratica, in analogia con la gerarchia di caches L1, L2 ed L3 in cui ogni livello è più grande e più lento del precedente, vogliamo creare una cache di secondo livello in cui ogni via è allo stesso tempo più grande e lenta della precedente. Il resto del lavoro è organizzato come segue. Il capitolo 2 presenta le caches NUCA, S-NUCA e D-NUCA e le differenze in termini di prestazioni che queste hanno fra loro. Il capitolo 3 definisce la cache TD-NUCA (ovvero Triangular Dynamic-Non Uniform Cache Access) e tutte le politiche di gestione che la riguardano. Gli strumenti usati per la valutazione delle performance sono discussi nella sezione 4. Nel capitolo 5 sono mostrate le valutazioni statistiche del progetto presentato, ed infine nel capitolo 6 saranno date le principali conclusioni.

*Capitolo 2***STATO DELL'ARTE**

Nel presente capitolo saranno esposte le tecniche note fino ad adesso. Inizieremo dalla valutazione delle caches UCA (Uniform Cache Access) per stabilire alcuni indici di riferimento utili nelle prove successive. Si procederà poi con l'esaminazione delle caches NUCA, ed in particolare delle Static-NUCA e delle Dynamic-NUCA, per poi riportare i principali risultati di valutazione delle stesse tenendo sempre presente la comparazione delle loro prestazioni. Infine saranno comparate alcune politiche di gestione tipiche delle caches D-NUCA. Nel presentare queste architetture di caches ci siamo spesso riferiti ai lavori di Keckler et al. [2], [6].

L'obiettivo di questa sezione è quindi arrivare a costruire caches a tempo di accesso non uniforme cercando anche particolari tecniche di gestione che riescano ad ottimizzare le prestazioni in termini di IPC e latenze medie.

2.1 UCA CACHES

Fino ad oggi sono usate, almeno in ambiti commerciali, più livelli di cache in tecnologia Uniform Cache Access ovvero a tempo di accesso costante ed indipendente dall'indirizzo del blocco oggetto dell'accesso stesso [11], [12], [13], [14]. Queste tradizionali caches sono suddivise in una matrice di banchi di memoria in modo da diminuirne la latenza di accesso complessiva, ma hanno un tempo di accesso calibrato su quello del banco più lontano dalla porta di ingresso della cache. L'accesso ai banchi più vicini perciò ha una latenza maggiore di quella teorica possibile. Inoltre soluzioni multiporta, che riescono ad evitare eventuali conflitti fra i vari accessi, e che permettono più accessi contemporanei, sono tecnologicamente più costose e meno performanti perché devono inevitabilmente aumentare l'area del chip e quindi non vengono adottate [4]. Seguendo le previsioni fatte dalla SIA (Semiconductor Industry Association [19], [15]) nel 2001 [3], per tutte le seguenti prove di simulazione abbiamo assunto una grandezza costante dell'area di silicio relativo alla cache L2 e, in considerazione delle tecnologie di costruzione dei transistor, saranno provate caches da: 2MB a 130nm (prevista dalla SIA per l'anno 2001), 4MB a 100nm (prevista per l'anno 2003), 8MB a 70nm (prevista per l'anno 2006) e 16MB a 50nm (prevista per l'anno 2009). Le caches provate sono associative a insiemi ed hanno una suddivisione in banchi compatibile con la roadmap SIA [3] (per ogni test di valutazione è riportata in tabella la suddivisione in banchi usata per le caches).

La seguente tabella riporta le prestazioni (in termini di IPC, miss rate, e latenze medie) di una cache UCA al variare della sua grandezza. Le caches provate nel lavoro [2] sono da 2MB, 4MB, 8MB, e 16MB come mostrato in tabella.

Tech (nm)	L2 Capacity	Num Sub-banks	Unloaded Latency	Loaded Latency	IPC	Miss Rate
130	2MB	16	13	67,7	0,41	0,23
100	4MB	16	18	91,1	0,39	0,20
70	8MB	32	26	144,2	0,34	0,17
50	16MB	32	41	255,1	0,26	0,13

Tabella 1: IPC e miss rate di una cache UCA al variare della sua dimensione.

L'unloaded latency è il tempo di accesso alla cache, corrispondente a quello del banco più lontano. Il loaded latency è il tempo di accesso alla cache in presenza di conflitti (è sperimentale).

La terza colonna indica la suddivisione in sotto-banchi di ogni banco della cache ed è ottenuta come migliore soluzione considerando le previsioni sulla grandezza del banco. La colonna unloaded latency riporta la media dei tempi di accesso (in cicli di clock) assumendo una distribuzione uniforme nei tempi di accesso dei banchi della cache (UCA) e assumendo nessun conflitto negli accessi. Questi valori non sono sperimentali ma sono stati trovati con strumenti di modellazione di circuiti tipo HSPICE e danno l'idea del tempo di accesso dei banchi in assenza di altre richieste contemporanee. La colonna loaded latency invece indica lo stesso valore (latenza media di accesso alla cache) ma in presenza di conflitti negli accessi ed è calcolata in modo sperimentale con l'ausilio di strumenti di simulazione. Si può notare come la latenza di accesso sia già alta in assenza di conflitti e che come questi possano diventare un serio problema per le prestazioni (in media un accesso ad una cache da 16MB impiega 255 cicli di clock). Un conflitto può essere causato da due accessi diretti allo stesso banco (bank contention) oppure da due accessi che devono usare la stessa connessione (channel contention). Per ogni cache di grandezza diversa è inoltre riportato l'IPC trovato con il simulatore e la percentuale di miss in L2. Come si vede l'IPC decresce molto al crescere della dimensione della cache proprio perché

cache di dimensioni maggiori sono più complesse (hanno un numero maggiore di banchi) e causano un innalzamento dei ritardi complessivi.

Si capisce quindi che questo tipo di architettura non sia applicabile per grandi cache o comunque per cache in cui la latenza dei collegamenti assuma un ruolo importante (wire-dominated caches). Questi risultati possono essere presi come punto di partenza per i successivi sviluppi.

2.2 NUCA CACHES

Invece di organizzare le caches in più livelli con latenze di accesso discrete, è possibile disegnarne una in cui ogni banco che la costituisce possiede una latenza propria appartenente ad un continuo di valori. Così, i banchi più vicini al controller della cache avranno una latenza di accesso minore, e quelli che si trovano più lontano avranno la latenza massima, ma non pregiudicheranno il comportamento di tutti gli altri. Ovviamente per permettere questo c'è bisogno di una rete di interconnessione che permetta ad ogni banco un accesso indipendente.

Con questa architettura i dati sono mappati in modo statico nei banchi ed i bit più significativi della parte indice degli indirizzi determinano il banco oggetto dell'accesso. Questo fa in modo che ogni banco sia indirizzabile indipendentemente dagli altri e che quindi sia calibrato in modo a se stante. Ogni banco preso in considerazione nella successiva simulazione è associativo a insiemi a 4 vie per formare una cache complessiva anch'essa a 4 vie. La ricerca di un dato in una NUCA è incrementale, cioè una volta calcolato il giusto indice, viene confrontato il tag del banco della via più vicina al controller (e quindi più veloce) e solo in caso di miss si procede verso il banco immediatamente più lontano (e quindi più lento).

Pensandoci bene, questa organizzazione riprende la normale gerarchia di caches L1, L2 e L3, ma creando un continuo di diverse vie in cui poter caricare un blocco. Inoltre il principale vantaggio di questi tipi di caches rispetto alle suddette caches multilivello è che al loro interno non c'è la ripetizione dei blocchi da un livello ai successivi, non vale cioè il principio di inclusione. Ciò permette un notevole risparmio di spazio in quanto non è presente ridondanza dei dati nelle diverse vie. È facile capire quindi che a parità di dimensione una cache NUCA può contenere più dati rispetto ad una equivalente UCA organizzata su più livelli.

Saranno ora presentate due particolari organizzazioni di cache NUCA con riguardo alla rete che connette internamente i banchi.

2.2.1 Private Channels S-NUCA

È possibile connettere i banchi di una cache NUCA al controller attraverso canali privati. Vi sarà quindi una connessione privata per i dati ed una per gli indirizzi, posta fra il controller della cache ed ogni banco (figura 1). È ovvio però che, mentre ogni banco può sfruttare la massima velocità possibile in quanto non sono possibili conflitti di canale (channel contention), si viene a creare un grande overhead in termini di area di silicio occupata dal chip che porta ad una limitazione nel numero di banchi costituenti la cache. Nel modellizzare i conflitti sui banchi (bank contention) della S-NUCA si sono scelti due approcci, uno conservativo secondo il quale si considera il ritardo introdotto dai latches che consentono richieste multiple, ed uno più aggressivo secondo il quale si ipotizza nullo questo tempo.

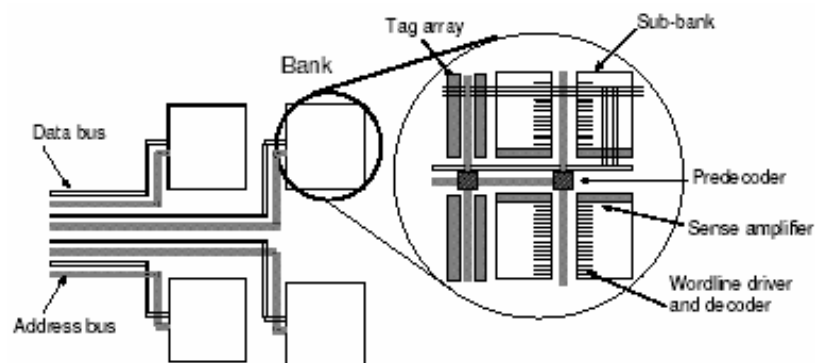


Figura 1: organizzazione interna di una cache Private Channels S-NUCA.

Ogni banco ha un canale privato che lo connette al controller. Si può vedere anche la suddivisione del banco in 4 sottobanchi.

La seguente tabella riporta i risultati dei test di simulazione di questa cache secondo i valori del lavoro [2]. L'ampiezza del canale di collegamento usato nelle prove è 128bit (per canale si intende il bus che collega direttamente il controller con ogni banco di cache).

Per ogni dimensione di cache è riportato in tabella il numero di banchi che la costituisce, la latenza di accesso di ogni banco, il tempo di accesso minimo alla cache, corrispondente all'accesso al banco più vicino, il tempo di accesso massimo, corrispondente all'accesso al banco più lontano ed il tempo di accesso medio, cioè la media dei tempi di accesso di ogni banco. Inoltre è riportato l'IPC trovato con il simulatore per i due diversi approcci e la latenza in presenza di carico (conflitti di banco).

Tech (nm)	L2 Capacity	Num banks	Unloaded Latency				Conservative		Aggressive	
			bank	min	max	avg	Load	IPC	Load	IPC
130	2MB	16	3	7	13	10	11,3	0,54	10,0	0,55
100	4MB	32	3	9	21	15	17,3	0,56	15,3	0,57
70	8MB	32	5	12	26	19	21,9	0,61	19,3	0,63
50	16MB	32	8	17	41	29	34,2	0,59	30,2	0,62

Tabella 2: IPC, unloaded latency e loaded latency di una Private Channels S-NUCA al variare della sua dimensione.

La latenza del banco è il tempo di accesso dello stesso. I tempi min, max e avg indicano il ritardo di accesso ai banchi più vicini, a quelli più lontani ed il tempo medio rispettivamente. L'approccio conservativo considera il ritardo introdotto dai latches che consentono le richieste multiple, l'approccio aggressivo non lo considera.

La differenza sostanziale tra questi risultati e quelli ottenuti per le caches UCA è che in questo caso l'IPC cresce, anche se di poco (7% in più di una cache da 8MB rispetto a una da 2MB), al crescere della dimensione e del numero di banchi della cache. Questo avviene proprio perché all'aumentare del numero delle vie (e quindi della dimensione) non c'è un corrispondente

aumento della latenza dei banchi delle vie precedenti che rimangono indipendenti le une dalle altre. In questa soluzione però la pesantezza dei collegamenti privati si fa sentire per caches di grandi dimensioni. Infatti la latenza massima, e quindi anche media, della cache da 16MB subisce un innalzamento molto più alto rispetto alle caches più piccole. Questo spiega perché l'IPC della cache da 16MB si riduce del 2% rispetto a quello della cache da 8MB.

Infine la pesantezza dei collegamenti privati si nota anche per quanto riguarda la migliore suddivisione in banchi della cache: si nota dalla terza colonna che il numero di banchi non va mai oltre 32. Questo avviene perché all'aumentare del numero dei banchi, i canali aumentano in numero ed in lunghezza, causando un aumento delle latenze di accesso ai singoli banchi. Perciò, per ottenere caches di dimensioni maggiori si deve necessariamente aumentare la dimensione dei banchi, causando ancora una volta un aumento dei tempi di risposta dei banchi stessi. Siamo quindi in presenza di un compromesso fra la dimensione massima della cache e la latenza dei suoi banchi.

2.2.2 Switched Channels S-NUCA

La seconda topologia di rete considerata (figura 2) usa una connessione mesh con doppi canali punto-punto da 128bit che collegano semplici routers situati presso ogni banco. I canali sono doppi in modo da consentire connessioni bidirezionali e limitare i conflitti. In questa implementazione c'è un ridotto uso di fili e per questo il numero di banchi adottabili cresce e con questo cresce anche la dimensione massima raggiungibile dalla cache. Come vedremo infatti dai risultati di test saranno le caches più grandi a beneficiare di questa implementazione. Il principale svantaggio di questa soluzione è dovuto alla maggiore complessità dei routers presso ogni banco.

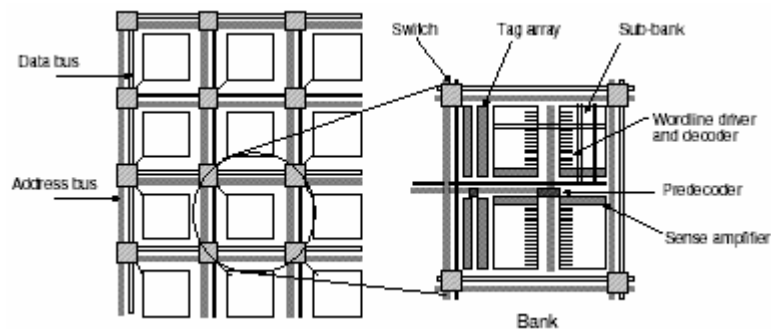


Figura 2: organizzazione interna di una cache Switched Channels S-NUCA.

Si nota la rete mesh fra i banchi: ogni banco ha un proprio switch connesso agli switch dei banchi vicini.

Nelle prove di test si è assunta l'ipotesi ottimistica che ogni router riesca a memorizzare le richieste con un buffer da 16 pacchetti in modo da non perdere i pacchetti oggetto dei conflitti di canale. La tabella 3 mostra i risultati della simulazione svolta nel lavoro [2]. L'estrema semplicità della rete permette una maggiore velocità negli accessi ai banchi di ogni tipo di

cache come si vede dalle latenze riportate in tabella (min, max e avg), ma è per le caches da 16MB che la latenza scende maggiormente, da 34,2 delle S-NUCA-1 a 24,2 della S-NUCA-2, portando un aumento dell'IPC di circa il 10% rispetto alle caches da 2MB (da 0.55 a 0.65).

Tech (nm)	L2 Capacity	Num banks	Unloaded Latency			Loaded Latency	IPC	Bank Req	
			bank	min	max				avg
130	2MB	16	3	4	11	8	9,7	0,55	17M
100	4MB	32	3	4	15	10	11,9	0,58	16M
70	8MB	32	5	6	29	18	20,6	0,62	15M
50	16MB	32	8	9	32	21	24,2	0,65	15M

Tabella 3: IPC e latenze di una Switched Channels S-NUCA al variare della sua dimensione.

La latenza del banco è il tempo di accesso dello stesso. I tempi min, max e avg indicano il ritardo di accesso ai banchi più vicini, a quelli più lontani ed il tempo medio. Sperimentalmente è calcolato l'IPC, la latenza in presenza di carico ed il numero di accessi ai banchi.

2.3 D-NUCA CACHES

I vantaggi maggiori dell'architettura NUCA si possono ottenere posizionando i dati più frequentemente acceduti nelle posizioni più veloci ed allontanando da queste i meno usati. In questa sezione mostreremo la migrazione dei dati dai bank più lontani al controller a quelli più vicini in modo dinamico e secondo politiche LRU. Di seguito saranno analizzate alcune politiche di gestione di questi tipi di caches con lo scopo di ridurre i tempi medi di accesso della L2 e di migliorare le prestazioni complessive. Per stabilire le politiche di migrazione si deve rispondere a tre domande: come i dati sono mappati nei bank, come è possibile ritrovare i dati, sotto quali condizioni i dati possono migrare.

2.3.1 Mappaggio dei blocchi

Dato l'alto numero di bank, potremmo mappare un blocco in modo statico in un solo bank (come per le S-NUCA), o, all'altro estremo, potremmo mappare il blocco in modo dinamico in qualsiasi bank. Un compromesso tra flessibilità di piazzamento dei dati e velocità nel ritrovarli fra i bank, è la soluzione a spread sets, in cui la matrice di bank che costituisce la cache è considerata come un struttura set-associative ed ogni bank corrisponde ad una via di un particolare insieme. La collezione dei bank che compongono un insieme è detta bank set e la sua cardinalità definisce l'associatività della cache. Ad esempio una cache associativa a 4 insiemi e composta da 32 bank, ha 8 bank sets di 4 bank l'uno.

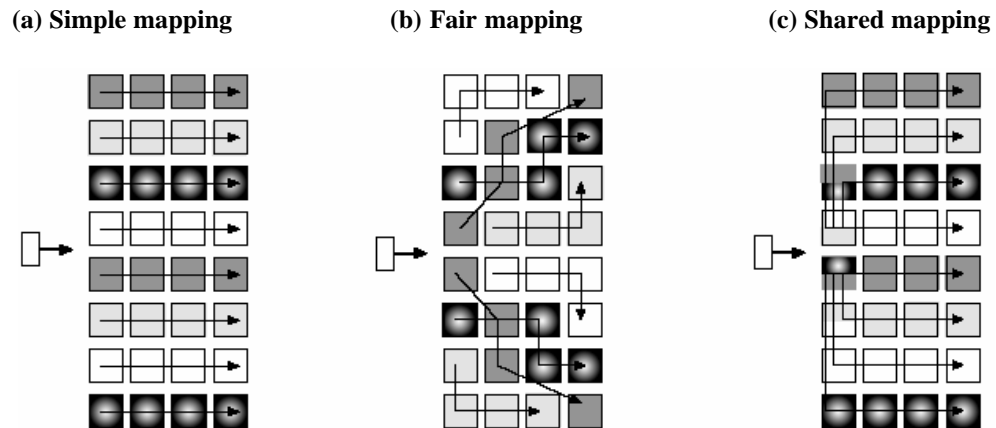


Figura 3: mappaggio dei bank sets nei banchi.

Un bank set è la collezione di banchi che costituisce un insieme. La sua cardinalità definisce l'associatività. I bank sets sono distribuiti sui banchi fisici nel modo indicato dalle frecce. Con il simple mapping i bank sets laterali saranno svantaggiati rispetto a quelli centrali. Con il fair mapping la disparità rimane considerevole a livello della prima via (primo banco) di ogni bank set. Nel fast shared mapping, i primi banchi (più veloci) sono condivisi da ogni bank set.

È possibile mappare i blocchi nei bank set in 3 modalità differenti. Nel simple mapping, mostrato in figura 3a, ogni riga di banchi assume il ruolo di bank set ed ogni banco che la compone appartiene ad una diversa via dello stesso insieme. La ricerca di un dato avviene quindi selezionando il particolare bank set, poi selezionando l'insieme giusto all'interno di questo ed infine comparando i tag via dopo via. Sono presenti però degli inconvenienti: primo, il numero di insiemi desiderati nella cache può non corrispondere con il numero delle righe che si vengono a creare con i bank set; secondo, i tempi di accesso dei bank sets non sono uniformi. Infatti, ipotizzando la porta di ingresso alla cache nel centro delle righe, i bank sets centrali avranno una latenza minore di quelli laterali. La seconda politica, la fair mapping (figura 3b), aggiunge un po' di complessità per risolvere i problemi sopra citati. Il mappaggio dei bank sets sui banchi fisici segue adesso le frecce mostrate in figura. Con questo modello il mappaggio è fatto

in modo che i tempi medi di accesso degli interi bank sets risultino simili. Proprio perché i bank sets non corrispondono più alle righe di banchi fisici è possibile variare il numero di banchi (e quindi di vie) che compongono i bank sets stessi. Però, anche con questa soluzione, i banchi più vicini al controller di ogni bank set (che saranno i più acceduti), non sono equidistanti al controller stesso e ciò può creare una disparità fra i vari insiemi (almeno a livello della prima via). L'ultima soluzione, la fast shared mapping di figura 3c, permette a tutti i bank sets di condividere i banchi più vicini al controller in modo da fornire la stessa latenza (quella minore possibile) ai banchi più usati di ogni insieme. Tutto ciò ovviamente comporta un certo overhead di complessità nella gestione dei banchi stessi.

2.3.2 Ricerca dei dati

Una volta selezionato il corretto bank set a partire dall'indirizzo, e selezionato il corretto insieme all'interno di questo, la ricerca del blocco fra le vie della cache prosegue secondo due possibili modalità: ricerca incrementale o multicast. La soluzione a ricerca incrementale consiste nella comparazione dei tag partendo dalla via più vicina al controller e proseguendo allontanandosi da questo solo in caso di miss della via considerata. È ovvio che la ricerca in ogni via deve prima attendere la terminazione della ricerca nella via precedente. Questa politica, dato che genera solo lo stretto numero necessario di messaggi di ricerca (o pacchetti di accesso) all'interno della rete di interconnessione, permette di tenere basso il consumo di energia, ma se il blocco dovesse trovarsi nelle ultime vie oppure non fosse presente in cache, si avrebbe una notevole riduzione delle prestazioni. Con la politica a ricerca multicast, i tag di tutte le vie vengono comparati in parallelo permettendo un notevole risparmio di tempo in caso di miss. Per attuare questa soluzione però, dal controller devono

essere diretti tanti messaggi di ricerca quanta è la cardinalità del bank set, ma solo uno di questi tornerà indietro con il risultato. Questa tecnica quindi consuma una maggiore quantità di energia rispetto al caso precedente, ed inoltre i troppi messaggi scambiati nella rete possono limitare leggermente le prestazioni a causa della maggiore possibilità di conflitti, sia di canale che di banco. Per questo sono state studiate ulteriori tecniche ibride: la limited multicast, secondo la quale solo le prime M vie delle N presenti sono utilizzate in parallelo per poi procedere in modo sequenziale; la partitioned multicast, nella quale le vie sono partizionate in sottoinsiemi che sono acceduti in parallelo fra loro con una ricerca incrementale al loro interno; la limited search, nella quale si parte con una ricerca incrementale fra le prime vie per poi finire in modo parallelo consentendo di limitare la latenza nel worst-case. È evidente che queste politiche sono un compromesso fra il tempo di accesso alla cache ed il consumo di energia. Le prestazioni di ognuna di queste saranno valutate nel seguito.

2.3.3 Migrazione dei blocchi

L'obiettivo della migrazione è quello di massimizzare gli hit nei bank più vicini e veloci avvicinando i blocchi maggiormente acceduti. È naturale quindi usare una politica di scambio dei blocchi basata sull'LRU. Una politica di questo tipo, che promuove ad ogni accesso i blocchi in posizioni più vicine, genera però molti movimenti di dati nella rete di interconnessione, aumentando il consumo di energia e la possibilità di conflitti. Per trovare un compromesso sono state studiate varie soluzioni.

In generale il problema della migrazione è risolto grazie alla definizione di tre diversi meccanismi che rispondono ai problemi di inserzione dei dati in cache, di espulsione e di promozione degli stessi. Per quanto riguarda l'inserzione di un blocco in seguito al verificarsi di una miss sullo stesso, è

possibile scegliere l'inserzione in coda alla lista delle vie (in pratica l'inserzione avviene nella via più lontana), in modo da evitare di allontanare blocchi più importanti e frequentemente acceduti. Così facendo però, si verrà a generare un maggior numero di scambi nel caso lo stesso blocco sia nuovamente acceduto in seguito. In alternativa è possibile inserire il nuovo blocco in testa o in una posizione casuale all'interno del bank set allontanando il blocco vittima dell'inserzione. È possibile inoltre limitare il numero delle copie dei blocchi da allontanare applicando una politica zero-copy, rimpiazzando cioè direttamente la vittima scelta per l'inserimento (potrebbe però essere eliminato un blocco importante nel caso di inserzione in testa o nel mezzo), o one-copy, copiando cioè la vittima dell'inserzione in una posizione più lontana e rimpiazzare dalla cache quest'ultimo blocco. Va da se che con una tecnica one-copy, il blocco rimpiazzato sarà poco importante, ma sarà generato uno scambio in più all'interno della cache con conseguente aumento del numero degli accessi ai banchi e quindi del consumo di energia e del numero dei possibili conflitti.

Per quanto riguarda l'espulsione dei blocchi è ritenuta ragionevole l'ipotesi di buttar fuori il blocco in ultima posizione (il meno recentemente usato) in quanto è quello meno importante fra quelli presenti in cache oppure quella di espellere un blocco in posizione casuale.

Le tecniche di rimpiazzamento da usare potranno essere allora l'inserzione in coda con una zero-copy ed una estrazione finale dalla coda, oppure una inserzione nel mezzo (o casuale) con una conseguente one-copy in coda e l'estrazione da questa, o infine una inserzione in testa, una successiva one-copy ed una estrazione casuale.

Infine, il protocollo di promozione è sostanzialmente uno solo al quale si può scegliere di applicare o meno qualche variante. La promozione viene detta generazionale perché propone di scambiare ogni blocco acceduto in cache con quello immediatamente più vicino al controller in seguito al verificarsi di un hit sul blocco stesso. Questa tecnica segue l'algoritmo

LRU, infatti i blocchi più acceduti migreranno velocemente nelle prime posizioni a scapito di quelli meno importanti che si allontaneranno da queste. Per limitare il numero di scambi effettuati, è possibile studiare varie euristiche variando il numero di hit necessari ad un blocco per effettuare una migrazione in avanti e variando la distanza (in numero di banche) che questo può coprire ad ogni promozione. Fra le tecniche valutate in seguito vi sono: 1 banco ogni hit, 1 banco ogni 2 hit, 2 banche ogni 2 hit e 2 banche ogni hit.

Per valutare le prestazioni di ogni tecnica sopra esposta, il lavoro [2] ha prima costruito una configurazione di base a cui in seguito fa riferimento nelle successive misurazioni delle politiche avanzate. Questa configurazione di base prevede le politiche più semplici fra quelle menzionate: simple mapping, ricerca multicast, promozione di un banco ad ogni hit e una politica di rimpiazzamento che espelle il blocco in coda ed inserisce il nuovo blocco al suo posto (zero-copy).

La tabella 4 mostra le prestazioni in termini di latenze, IPC, e tasso di miss della D-NUCA in configurazione base. Nella terza colonna è riportata l'organizzazione della matrice di banche di ogni dimensione di cache: la cache da 2MB è composta da 4 bank sets ognuno di 4 vie, la cache da 4MB ha 8 bank sets da 4 vie, la cache da 8MB ha 16 bank sets da 8 vie ed infine la cache da 16MB ha 16 bank sets da 16 vie. Nel secondo blocco della tabella si può leggere il tempo di accesso di ogni banco, la latenza di accesso minima (latenza per l'accesso al banco più vicino), la latenza di accesso massima (latenza per l'accesso al banco più lontano) e la latenza media di accesso, tutte considerate in assenza di conflitti. Si nota che a differenza delle S-NUCA 1, le latenze massime sono estremamente basse considerato l'elevato numero di banche che compone la cache. La cache da 16MB infatti è composta da ben 256 banche, ma nonostante ciò ha una latenza massima di soli 47 cicli di clock (una S-NUCA 1 da 16MB è composta da soli 32 banche ed ha una latenza massima di 41 cicli). Questo

permette di poter aumentare più liberamente il numero di banchi riducendone la dimensione e quindi il tempo di accesso introdotto da ognuno di essi. Si nota infatti che il tempo di accesso dei banchi anche per le grandi caches è di soli 3 cicli, mentre per soluzioni S-NUCA (sia in versione 1 che 2), che presentano banchi di dimensioni maggiori, questo ritardo cresce fino a 8 cicli. Per quanto riguarda le prestazioni si osserva che per le caches di più piccole dimensioni (4MB) si ha un leggero miglioramento (circa il 2%) sulle prestazioni delle S-NUCA più performanti, mentre per dimensioni maggiori (16MB) il miglioramento può arrivare fino al 6%.

Lo svantaggio principale di queste caches riguarda l'elevato numero di messaggi che circolano all'interno della rete di interconnessione ed il conseguente alto numero di accessi ai banchi (266M contro 15M per cache da 16MB) anche in conseguenza del fatto che per questa cache si è scelta la ricerca in modalità multicast. Tutto ciò aumenta il consumo energetico del dispositivo ed aumenta la probabilità di conflitti fra due o più accessi contemporanei. Si deve considerare però che ogni banco è molto più piccolo rispetto al caso S-NUCA (quindi consumerà meno) e che il loro numero è molto più grande.

Tech (nm)	L2 Cap.	Bank org.	Unloaded Latency			Load avg	IPC	Miss rate	Bank Req	
			bank	min	max					avg
130	2MB	4x4	3	4	11	8	8,4	0,57	0,23	73M
100	4MB	8x4	3	4	15	10	10,0	0,63	0,19	72M
70	8MB	16x8	3	4	31	18	15,2	0,67	0,15	138M
50	16MB	16x16	3	3	47	25	18,3	0,71	0,11	266M

Tabella 4: latenza media, IPC e miss rate di una DNUCA al variare della sua dimensione.

È riportata l'organizzazione della matrice di banchi (bank org), le latenze in assenza di conflitti (unloaded latency), le latenze con carico, l'IPC, il miss rate, ed il numero di accessi ai banchi.

La tabella 5 mostra gli effetti della calibrazione della D-NUCA da 16MB sulle varie politiche. È risultato dalle prove che una ricerca incrementale divide per tre il numero degli accessi ai banchi rispetto alla ricerca multicast (89M contro 266M) ma presenta un IPC ridotto del 6%. Politiche di ricerca ibride riescono a ridurre molto il numero degli accessi ai banchi (riducendo i conflitti) senza però penalizzare troppo le prestazioni. Queste soluzioni sono da tenere in considerazione in previsione di dispositivi nei quali il consumo energetico è molto rilevante. Una ottima soluzione, come compromesso fra prestazioni e risparmio energetico, è risultata quella della ricerca multicast fra le prime 2 vie e incrementale per le successive. Questo risultato conferma che la quasi totalità degli hit alla L2 avviene nelle prime 2 vie, e proprio per questo la ricerca incrementale sulle successive permette di ridurre del 50% il numero degli accessi ai banchi.

Cambiando la funzione di mapping da simple a fast shared, le prestazioni generali risultano aumentate, ma siamo in presenza di una maggiore complessità dei routers, dei banchi e del controller.

Le diverse politiche di promozione non incidono molto sulle prestazioni come si vede dai tre esperimenti fatti, anche se le soluzioni di promozione ogni 2 hit fanno decrescere leggermente il numero di accessi ai banchi da 266M della soluzione di base a 259M.

Infine la tecnica migliore per il rimpiazzamento è risultata quella di inserire il nuovo blocco in testa alle vie, copiare il blocco rimpiazzato in una via (in modo casuale) più lontana ed eliminare il blocco trovato in quest'ultima. Questa soluzione presenta una latenza media minore (15.5 cicli) ma per contro causa un maggior numero di accessi ai banchi (267M).

Policy	Avg Lat	IPC	Miss rate	Bank Acc
Search				
incremental	24,9	0,65	0,114	89M
2mcast + 6inc	23,8	0,65	0,113	96M
2inc + 6mcast	20,1	0,70	0,114	127M
2mcast + 6mcast	19,1	0,71	0,113	134M
Mapping				
fast shared	16,6	0,73	0,119	266M
Promotion				
1bank/2hit	18,5	0,71	0,115	259M
2bank/1hit	17,7	0,71	0,114	266M
2bank/2hit	18,3	0,71	0,115	259M
Eviction				
insert tail, evict random, 1copy	15,5	0,70	0,117	267M
insert middle, evict random, 1copy	16,6	0,70	0,114	267M
Baseline				
simple map, mcast, 1bank/1hit, insert tail	18,3	0,71	0,114	266M

Tabella 5: valutazione delle singole politiche avanzate della D-NUCA.

La ricerca incrementale riduce ad un terzo il numero degli accessi ai banchi diminuendo l'IPC del 6%. Una ottima soluzione è la ricerca multicast fra le prime 2 vie ed incrementale per le successive la quale dimezza il numero degli accessi mantenendo pressoché invariate le prestazioni.

Il fast shared mapping non produce sostanziali miglioramenti rispetto alla configurazione base soprattutto se si considera anche la maggiore complessità che è richiesta ad ogni dispositivo (banchi, controller e routers).

Le politiche di promozione ogni 2 hit diminuiscono sensibilmente il numero degli accessi senza variare le prestazioni.

La tecnica di rimpiazzamento migliore è risultata quella di inserzione in testa, con una one-copy in una posizione random da espellere dalla cache. La latenza media è ridotta del 3% con un mero aumento nel numero degli accessi ai banchi.

Un'ultima osservazione da notare comparando una organizzazione UCA multilivello provvista delle caches di secondo e terzo livello, con una cache L2 D-NUCA, è il notevole risparmio di area di silicio da parte della tecnica sperimentata. Infatti con una cache D-NUCA della stessa grandezza della

L3, si riescono ad ottenere prestazioni migliori (la differenza è accentuata per grandi dimensioni di caches L3) rispetto al caso multilivello, con in più il risparmio dello spazio occupato dalla cache L2 che non è presente (o meglio, che è sostituita dalla D-NUCA).

*Capitolo 3***CACHES D-NUCA TRIANGOLARI**

Obiettivo del nostro studio è quello di trovare una organizzazione della cache D-NUCA tale per cui a parità di area di silicio occupata, fornisca prestazioni più elevate, o, equivalentemente, con un'area di silicio fortemente ridotta non degradi troppo i suoi tempi di risposta. Si cercherà quindi un compromesso fra risparmio nello spazio di silicio e prestazioni. Notare che risparmiare sulla superficie del chip porta ad una minore complessità dei circuiti e quindi sia ad un minor consumo energetico che ad una migliore dissipazione del calore.

Nelle sezioni seguenti sarà definita la struttura triangolare e le relative tecniche di gestione. Successivamente sarà valutata l'opportunità di allungare gradatamente i tempi di risposta dei banchi di ciascuna via, in modo da ridurre ulteriormente il consumo di energia da parte della cache.

3.1 DEFINIZIONE

Una volta definita e valutata l'architettura della cache Dynamic-Non Uniform Cache Access, procediamo ora ad introdurre una sostanziale innovazione. In analogia con la gerarchia di caches UCA multilivello, immaginiamo di avere una cache D-NUCA con una struttura composta non da una matrice di banchi ma più complessa, cioè con una dimensione di pagina di memoria che va crescendo all'allontanarsi dal controller. Se pensiamo infatti alle caches UCA multilivello, è presente una L1 piccola ma molto veloce, poi una L2 un po' più grande (e che contiene interamente la L1) ma più lenta, ed infine la L3 ancora più grande e più lenta. È possibile allora definire una cache D-NUCA in cui la prima via è grande poco più di una L1 ed ha una distanza minima dal controller, affiancargli poi una seconda via di ampiezza doppia (la pagina di memoria che può contenere ha quindi ampiezza doppia) ed un po' più lontana dal controller, e così facendo fino all'ultima via che dovrà avere una dimensione pari alla dimensione della pagina di memoria di una cache L3, in modo da permettere il caricamento nella cache di una zona di memoria della stessa grandezza. In figura 4 è mostrata l'organizzazione dei banchi di una cache D-NUCA rettangolare da 8x4 banchi comparata con la sua equivalente cache D-NUCA Triangolare. Come è facilmente intuibile una TD-NUCA occupa solo una parte dell'area di silicio impiegata da una normale D-NUCA, infatti come mostra la figura sono presenti solo 15 banchi contro i 32 della D-NUCA. In generale, per una cache rettangolare con n vie in cui la grandezza di pagina vale per ipotesi 2^{n-1} (in modo da consentire una corretta triangolarizzazione), si ha un numero totale di banchi dato da $n \cdot 2^{n-1}$. Nella equivalente cache triangolare il numero di banchi utilizzati, ipotizzando che ogni via sia di dimensioni doppie rispetto alla precedente, è dato dalla formula $\sum_{i=0}^{n-1} (2^i)$ corrispondente al valore $2^n - 1$. Il rapporto di questi

due valori al tendere di $n \rightarrow \infty$ tende a n , vale a dire che per un alto numero di vie il risparmio del numero di banchi delle caches triangolari è lineare nel numero di vie.

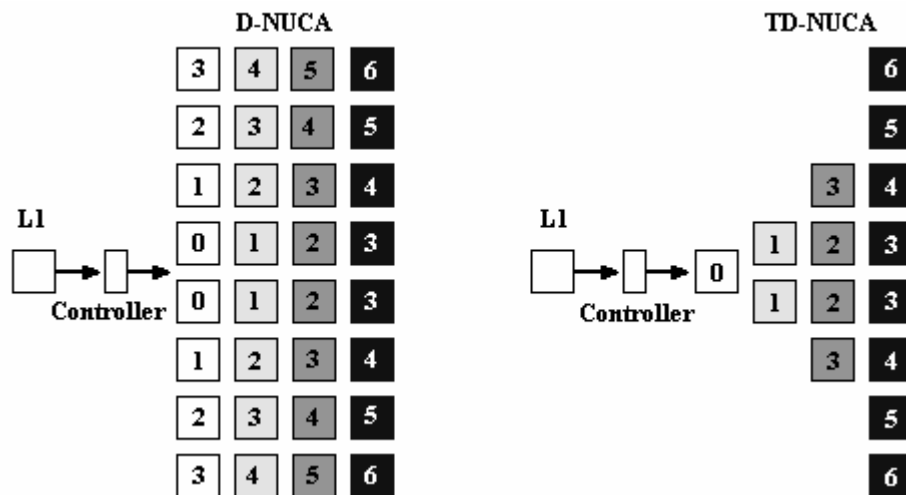


Figura 4: struttura dei banchi di una cache D-NUCA Triangolare a confronto con una D-NUCA rettangolare.

La TD-NUCA ha solo 15 banchi contro i 32 della normale D-NUCA con un risparmio del 50% sull'area di silicio del chip.

Ogni banco ha riportata la latenza che impiega un pacchetto di accesso per arrivarvi partendo dal controller (tempo di routing). Come si nota la "triangolarizzazione" di una cache rettangolare non è altro che un taglio effettuato su di questa, sia per quanto riguarda la struttura dei banchi, sia per i relativi tempi di routing.

Ipotizzando stesse latenze di accesso per le vie della D-NUCA e le corrispondenti vie della TD-NUCA è facile aspettarsi prestazioni molto inferiori dalla TD-NUCA in quanto il miss rate dovrà per forza aumentare rispetto all'equivalente rettangolare. Ma è poi vero che le latenze medie di accesso ai banchi rimangono le stesse? In figura 4 è riportato dentro ogni banco il tempo di routing che un messaggio (o pacchetto) di accesso al banco stesso impiega per arrivarvi, ipotizzando che un pacchetto si possa spostare di un banco (in orizzontale o in verticale) ad ogni ciclo di clock.

Come si vede la “triangolarizzazione” di una cache non è altro che un taglio effettuato sulla cache rettangolare in modo da scartare i banchi che rimangono fuori. Anche i tempi di routing dei banchi rimangono pertanto gli stessi. Quello che però cambia fra le due architetture è il posizionarsi dei singoli blocchi nei banchi. Supponiamo cioè di prendere un blocco il cui indirizzo è tale per cui debba essere posizionato nel primo bank set della cache rettangolare (il seguente esempio è mostrato in figura 5): i tempi di routing possibili per la ricerca di questo blocco sono 3 (se il blocco dovesse trovarsi nella prima via), 4 (se il blocco dovesse trovarsi nella seconda via), 5 (nella terza via) e 6 (nella quarta), con una media, in caso di hit per la ricerca, di 4.5 cicli di clock. Consideriamo ora lo stesso blocco durante una ricerca nella cache triangolare: i tempi di routing possibili sono adesso di 0 (nel caso si trovasse nella prima via), 1 (nel caso si trovasse nella seconda via), 3 (se si trovasse nella terza via) e 6 (se si trovasse nella quarta via), con una media di 2.5 cicli di clock.

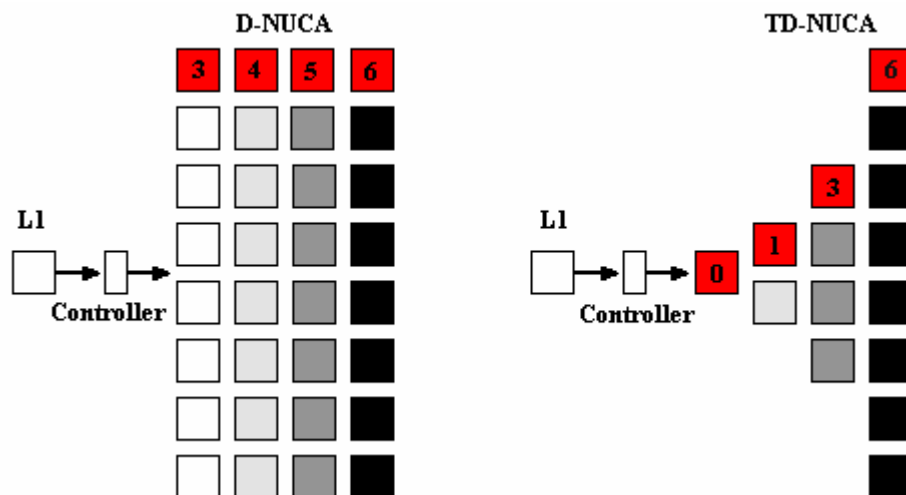


Figura 5: posizionamento di uno stesso blocco nelle 4 vie di una cache D-NUCA 8x4, e nelle 4 vie di una cache TD-NUCA [1,2,4,8].

Si può notare che in una cache triangolare il blocco risulta mediamente più vicino al controller fornendo tempi di accesso tipici di una cache di più piccole dimensioni.

È evidente quindi che questa cache ha i tempi di routing, e quindi di accesso, tipici di una cache di piccole dimensioni, ma permette di caricare nelle vie più lontane pagine di memoria comunque di grandi dimensioni. Quello che dovremo aspettarci perciò è, sì un aumento del tasso di miss (la dimensione della cache è comunque minore), ma anche una diminuzione dei tempi di accesso ai bank. Per maggiori informazioni su come è possibile posizionare i blocchi nei bank vedere i paragrafi seguenti.

Osserviamo ora la suddivisione dell'indirizzo (logico o fisico che sia). Essendo il numero di bank in ogni via non costante, è ovvio che la parte dell'indirizzo che indica l'indice del bank in cui posizionare un particolare blocco deve per forza di cose essere a grandezza variabile. La figura 6 mostra per una cache da 4 vie ognuna delle quali composta da 1, 2, 4 e 8 bank rispettivamente, la suddivisione dell'indirizzo e la posizione del campo utilizzato per fare la scelta sul posizionamento del bank (campo bank). È considerato uno spazio di indirizzamento a 32bit, la dimensione dei blocchi vale 64Byte ed i bank contengono 4096 insiemi. La prima via è composta da 1 bank e comprende 4096 insiemi, la parte indice dell'indirizzo usato per accedervi sarà allora di 12bit e non sarà presente il campo bank. La seconda via è composta da 2 bank e comprende $2 \cdot 4096$ insiemi, la grandezza dell'indice dovrà allora essere di 13bit che considereremo suddivisi nei 12bit meno significativi costituenti il campo index e nel bit più significativo che permette la scelta del bank in cui cercare il blocco. Continuando a percorrere le vie, ad ogni passo il bit meno significativo del campo tag passerà a costituire il campo bank dell'indirizzo che aumenterà di una unità alla volta. Questi concetti saranno approfonditi nella seguente sezione in cui ci si porrà il problema del posizionamento dei blocchi nei bank, ovvero del problema del mapping.

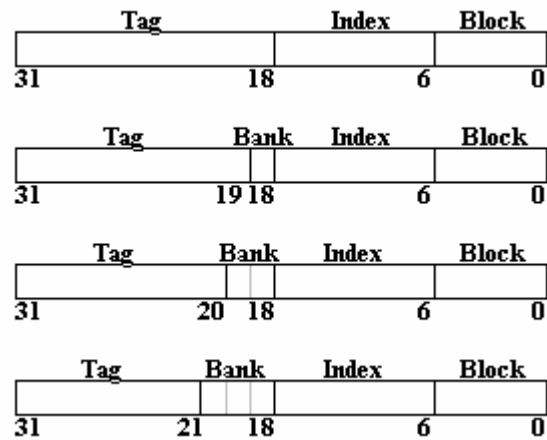


Figura 6: suddivisione dell'indirizzo di un byte per una cache TD-NUCA.

È considerato uno spazio di indirizzamento a 32bit, la dimensione dei blocchi vale 64Byte, i banchi contengono 4096 insiemi e sono presenti 4 vie rispettivamente composte da 1, 2, 4 e 8 banchi dalla prima all'ultima. Il campo tag assorbe i rimanenti bit più significativi dell'indirizzo.

L'accesso alla prima via non necessita della parte bank dell'indirizzo in quanto è presente un solo banco (prima suddivisione). L'accesso alla seconda via deve considerare 1bit per la scelta del banco in cui è posizionato il blocco, sottraendolo alla parte meno significativa della parte tag (seconda suddivisione). L'accesso alla terza via utilizza 2bit del campo bank diminuendo ulteriormente la grandezza del tag (terza suddivisione). L'accesso all'ultima via necessita di 3bit per la scelta del banco e lascia solo 11bit al campo tag (quarta suddivisione).

3.2 MAPPAGGIO DEI BLOCCHI

Veniamo ora a definire come ogni blocco può essere mappato nei bank di ogni via. La funzione che definisce il mapping è adesso più complicata perché al raddoppiare della dimensione della pagina di memoria da una via alla successiva, si deve considerare la doppia possibilità che ha un blocco di posizionarsi nella via successiva. Si verrà così a creare un albero binario di possibili posizioni che può assumere un blocco con un particolare indirizzo dalla prima via all'ultima. Definire le posizioni di un blocco in ogni via e quindi definire la forma dell'albero significa perciò definire la funzione di mapping. Notare che tutti i bank set avranno alcuni bank (i primi) in comune fra loro.

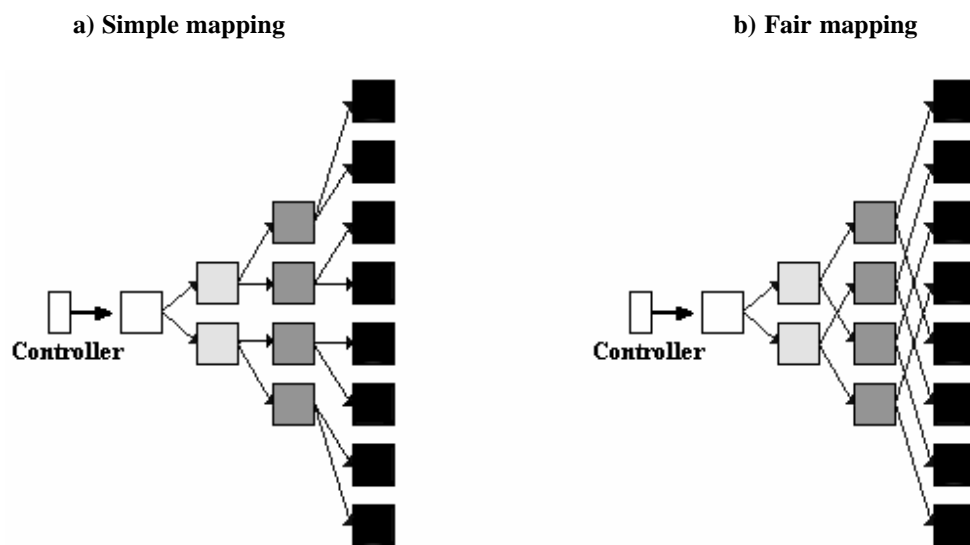


Figura 7: possibili mapping dei blocchi nei bank.

Nel simple mapping la funzione di mapping produce l'albero binario mostrato a sinistra. Si nota che i bank set laterali hanno tempi medi di accesso molto differenti da quelli più centrali. Nel fair mapping, con l'albero binario di destra, si ha la minima varianza fra i tempi medi di accesso di ogni bank set.

Si deve osservare inoltre che la funzione che calcola il mapping non deve essere complicata ma deve essere eseguita dal controller in un singolo ciclo di clock in modo da non introdurre ulteriori latenze nella ricerca di un blocco.

Abbiamo studiato due possibili mapping in analogia con gli equivalenti del caso D-NUCA. Nel simple mapping, mostrato in figura 7a, un blocco in posizione i di una via si può mappare nella via successiva (se ha dimensione doppia) nelle posizioni $i*2$ e $i*2+k$ (dove k è la dimensione del banco). Questo mapping però, analogamente al caso D-NUCA, penalizza i blocchi i cui indirizzi li portano ad essere caricati nei banchi laterali che avranno una latenza media di accesso più alta. I bank set laterali vengono quindi penalizzati maggiormente rispetto a quelli centrali. In termini implementativi è possibile realizzare questo mapping aggiungendo ad ogni via un bit nella parte bank dell'indirizzo (con riferimento alla figura 8a) e considerando quest'ultimo bit come il meno significativo del campo bank. Il campo bank dovrà quindi essere letto da destra verso sinistra.

Nel fair mapping, mostrato in figura 7b, la funzione che definisce la posizione dei blocchi nei banchi impone che un blocco in posizione i di una certa via si debba mappare nella via successiva (se ha dimensione doppia) nelle posizioni i e $i+k*n$ (dove n è il numero di banchi della via precedente). Questa tecnica di mapping porta ad una stessa media dei tempi di accesso di qualsiasi coppia di banchi di una stessa via che si riferiscono allo stesso blocco della via precedente secondo l'albero binario. Ad esempio, le latenze delle coppie di banchi della quarta via riferite allo stesso blocco della via precedente, secondo i valori delle latenze di routing di figura 4 sono: 6 e 3, poi 5 e 4, poi 4 e 5 ed infine 3 e 6; come si vede la media (4.5) rimane costante. In altre parole è possibile dire che la varianza (ovvero lo scostamento) dei tempi medi di accesso di ogni bank set dalla media dei tempi medi di accesso di tutti i bank set, è il minimo possibile. In termini implementativi questa funzione è realizzata aggiungendo alla parte bank

dell'indirizzo un bit per ogni via e considerando quest'ultimo come il meno significativo (figura 8b). Il campo bank si verrà quindi a leggere normalmente da sinistra verso destra.

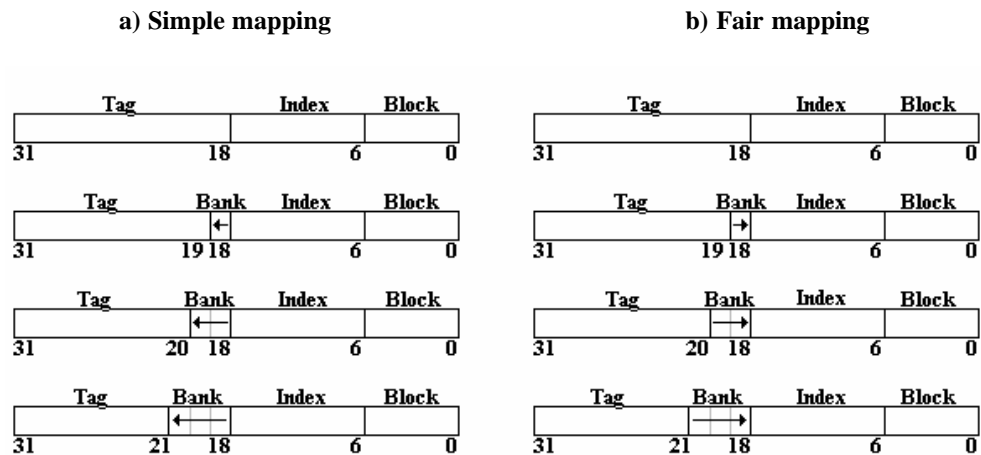


Figura 8: suddivisione dell'indirizzo e verso di lettura del campo bank nei due tipi di mapping.

Con il simple mapping il campo bank ha i bit più significativi a destra mentre quelli più a sinistra, sottratti al campo tag, sono considerati i meno significativi. Per implementare invece il fair mapping il campo bank deve essere letto normalmente da sinistra verso destra.

Come abbiamo visto il fair mapping è più equo del simple mapping in considerazione del fatto che ogni bank set ha un tempo di routing più vicino alla media totale.

Lo svantaggio principale del fair mapping rispetto al simple mapping è dovuto al diverso routing che si rende necessario per la distribuzione dei pacchetti di accesso ai banchi obiettivi della ricerca di un blocco. Come vedremo in seguito, il fair mapping genera un maggior numero di pacchetti di accesso in cache aumentando la probabilità di conflitti di canale (i conflitti di banco rimangono gli stessi in quanto il numero di banchi

acceduti nei due diversi mapping rimane costante). Per una trattazione più precisa sulle modalità di routing si rimanda alla sezione 3.4.

3.3 MAPPAGGIO VERTICALE DEI BLOCCHI

La distribuzione degli indirizzi consecutivi nei diversi banchi di una stessa via può essere fatta seguendo la modalità interlacciata o meno (notare che il problema del mapping della sezione precedente risponde alla necessità di posizionare i blocchi in ogni via, quindi distribuisce gli indirizzi in modo orizzontale sulla matrice dei banchi; in questa sezione invece vogliamo distribuire gli indirizzi nei banchi all'interno di una stessa via, quindi in modo verticale). La maniera più semplice di distribuire lo spazio degli indirizzi in una colonna di banchi relativi alla stessa via, supponiamo di due elementi, è quella di posizionare la prima metà nel primo banco e la seconda metà in quello successivo, secondo una distribuzione in modalità non-interlacciata (in figura 9a si suppone che ogni banco possa contenere k insiemi). Il difetto di questa modalità di mapping è che 2 indirizzi consecutivi sono contenuti nello stesso banco e se questi fossero acceduti in sequenza (come spesso avviene per programmi che rispettano il principio di località), il secondo dei due troverebbe molti conflitti (sia di canale che di banco) durante la ricerca del blocco in cache, peggiorando le prestazioni. Questa tecnica ha pertanto prestazioni più basse rispetto alla modalità interlacciata (come vedremo in seguito), e ci tornerà utile soltanto per validare il nostro simulatore TD-NUCA Sim-Alpha rispetto all'originale ([16], [17] e [18]). Per implementare questa modalità è sufficiente scegliere il banco in cui posizionare il blocco a seconda dei bit più significativi dell'indice (esteso) del blocco, cioè a seconda del valore del campo bank (letto nel giusto verso a seconda della modalità di mapping orizzontale).

La seconda modalità studiata è quella che ha la distribuzione degli indirizzi mostrata in figura 9b ed è usata nelle caches D-NUCA del lavoro [2]. Questa modalità si dice interlacciata in quanto pone qualsiasi 2 indirizzi consecutivi in due banchi diversi. Proprio per questo motivo le prestazioni misurate con tale tecnica risultano maggiori in conseguenza ad un minor numero di conflitti fra gli accessi ai banchi. L'implementazione di questa modalità di interlacciamento prevede che la scelta del banco in cui posizionare un blocco venga fatta in base al valore assunto dai bit meno significativi del campo indice dell'indirizzo. In pratica il campo bank viene spostato alla destra del campo indice.

Cercando di implementare questa modalità per le caches triangolari, in cui la dimensione dell'indice aumenta nello spostarsi fra le vie, ci siamo scontrati nel problema dello shift dell'index. Dovendo infatti posizionare il campo bank a destra del campo index, ed essendo il campo bank a dimensione crescente, ad ogni via l'indice deve subire uno shift verso sinistra per far posto a destra al campo suddetto. In pratica avremmo avuto un index che, shiftando verso sinistra, cambia valore ad ogni via provocando salti da un set all'altro da parte di uno stesso blocco nello spostarsi all'interno del proprio bank set. Abbiamo perciò provveduto a creare una seconda distribuzione degli indirizzi anch'essa interlacciata, come si può vedere dalla figura 9c (2 indirizzi consecutivi sono sempre in due banchi diversi), ma in cui il campo bank rimanga a sinistra del campo index. Per implementare questa distribuzione, cioè definire il modo di scegliere il banco in cui posizionare un blocco, viene effettuata la somma fra il campo bank e gli n bit meno significativi del campo index, dove n è la cardinalità del campo bank in tale via. La figura 10 chiarifica la funzione di mapping nelle combinazioni dei mapping orizzontali con quelli verticali e mostra la distribuzione degli indirizzi di ciascuna modalità. Dalle parti *b* e *d* della figura si nota come per applicare la modalità interlacciata, la lettura del

campo bank debba essere effettuata considerando la giusta direzione dello stesso, in accordo con il mapping orizzontale scelto.

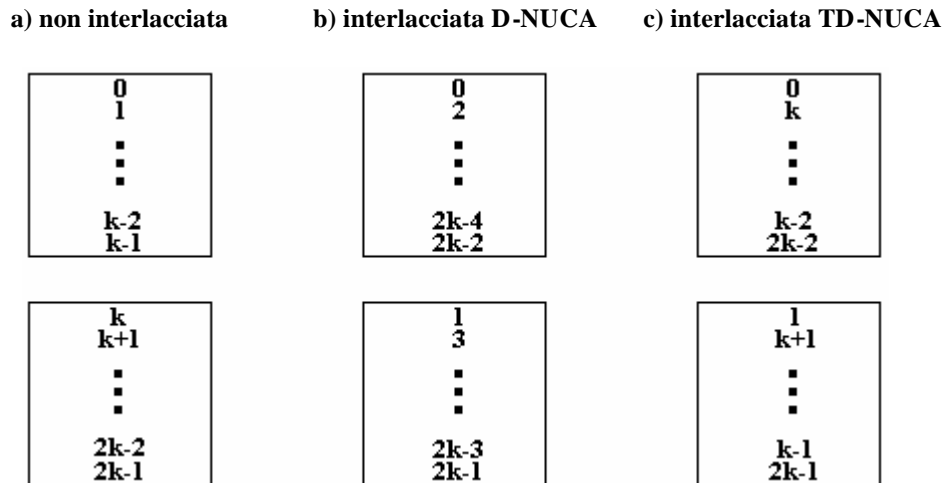


Figura 9: modalità di distribuzione degli indirizzi in una via composta da due banche.

Nella modalità non-interlacciata, 2 indirizzi consecutivi sono posizionati nello stesso banco. A causa di una maggiore probabilità di conflitti fra due accessi relativi ad indirizzi consecutivi ci aspetteremo prestazioni peggiori.

Nelle modalità interlacciate, 2 indirizzi consecutivi sono sempre posizionati in banche diversi. Si nota come siano state scelte due modalità interlacciate diverse per i due tipi di cache.

È evidente che data la diversità delle due modalità di interlacciamento scelte nel caso D-NUCA e TD-NUCA non sarà possibile ottenere risultati perfettamente uguali tra i due simulatori (D-NUCA Sim-Alpha e TD-NUCA Sim-Alpha) anche per analoghe configurazioni di cache. Per questo motivo è stato scelto di validare il simulatore TD-NUCA Sim-Alpha usando sia la modalità interlacciata che la modalità non-interlacciata, ma quest'ultima verrà poi abbandonata nelle prove che hanno l'obiettivo di migliorare le prestazioni.

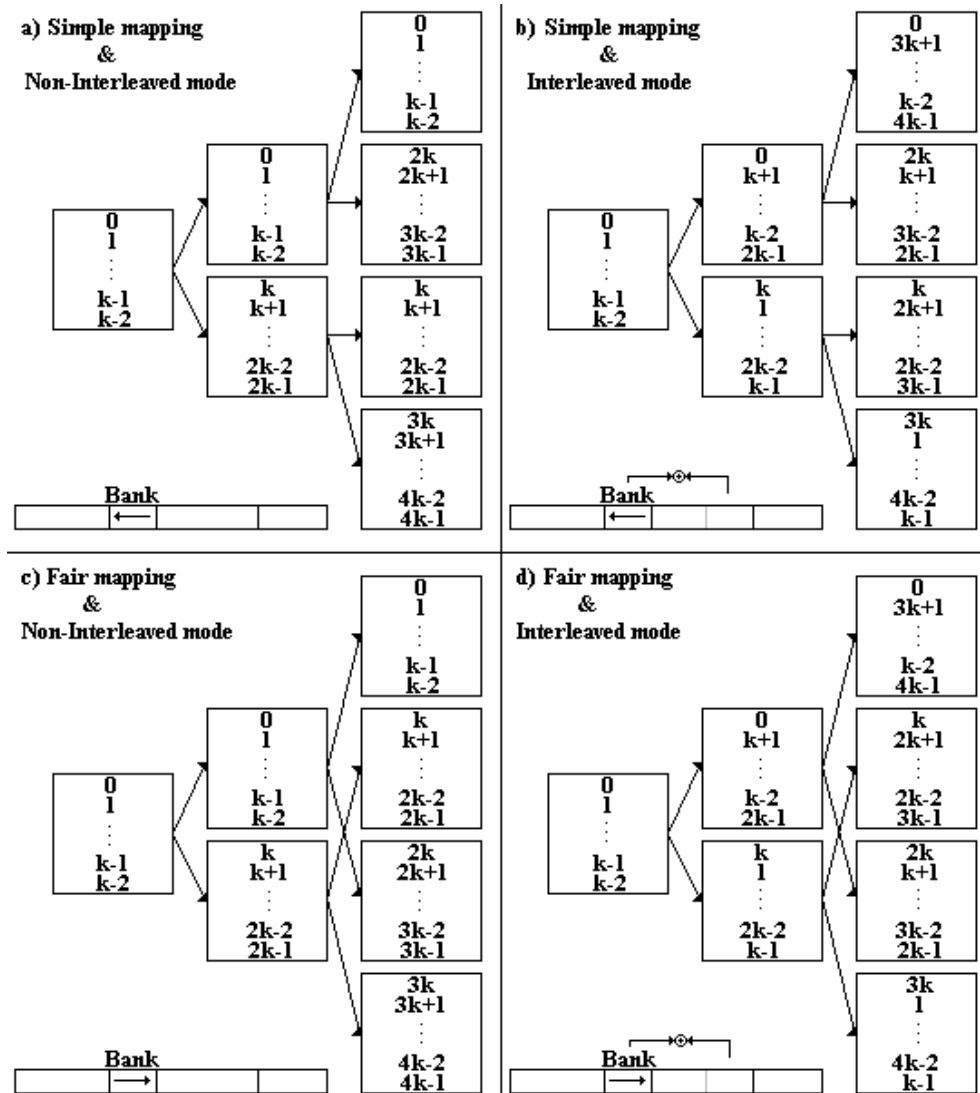


Figura 10: combinazioni tra le modalità di mapping orizzontale con quello verticale. Per ognuna di esse è mostrata la distribuzione degli indirizzi nei banche, e la funzione usata per effettuare la scelta del banco.

La differenza fra simple mapping e fair mapping si riversa nella differente direzione di lettura del campo bank.

Nella modalità interlacciata la scelta del banco prevede una preliminare somma fra il campo bank e la parte meno significativa del campo index.

3.4 ROUTING

Avevamo accennato in precedenza al problema del routing dei pacchetti di accesso alla cache. Il routing definisce il percorso intrapreso da ogni pacchetto di accesso per arrivare ai banchi obiettivi di ogni operazione di cache (lettura, scrittura, promozione...). In una cache D-NUCA i pacchetti di accesso seguono i canali mostrati in grigio in figura 11a spostandosi prima verticalmente fino ad arrivare alla giusta riga del banco (e del bank set) e poi orizzontalmente fino a raggiungere la colonna corretta. Durante la ricerca di un blocco in un bank set ad esempio, ci si sposterà sulla giusta riga e poi, a distanza di 1 ciclo di clock l'uno dall'altro (per ipotesi), si accederà ad ogni banco dalla prima via all'ultima sfruttando i minimi tempi possibili per arrivare in ogni banco. I canali che si rendono necessari a questo routing (detto routing 1) sono quelli mostrati in figura.

Volendo estendere questo tipo di routing alle caches triangolari, devono essere aggiunti dei canali verticali anche alle vie successive alla prima in quanto è l'unico modo, quello verticale, di raggiungere i banchi più esterni delle vie più grandi. Il routing 1 per le caches triangolari quindi impone che di via in via si debba raggiungere prima la riga metà della ricerca e poi la colonna (figura 11b), in modo analogo a quanto avviene per le caches rettangolari. Così facendo, i tempi di routing di ogni banco sono gli stessi dei corrispondenti nella cache D-NUCA, cioè sono i minimi possibili. Con questo routing un pacchetto di accesso per la ricerca di un blocco passa per ogni banco del bank set, vi accede, e poi prosegue nella via successiva. Questo è possibile perché abbiamo implicitamente supposto di utilizzare il simple mapping in modo da avere banchi consecutivi del bank set fisicamente vicini tra loro. Si può ad esempio notare come i due banchi più in alto della quarta via in figura 11b siano acceduti dal banco a loro più

vicino della via precedente cioè quello più in alto della terza via. Questo permette di accedere ai banchi con i tempi minimi.

a) Routing 1 in D-NUCA b) Routing 1 in TD-NUCA c) Routing 2 in TD-NUCA

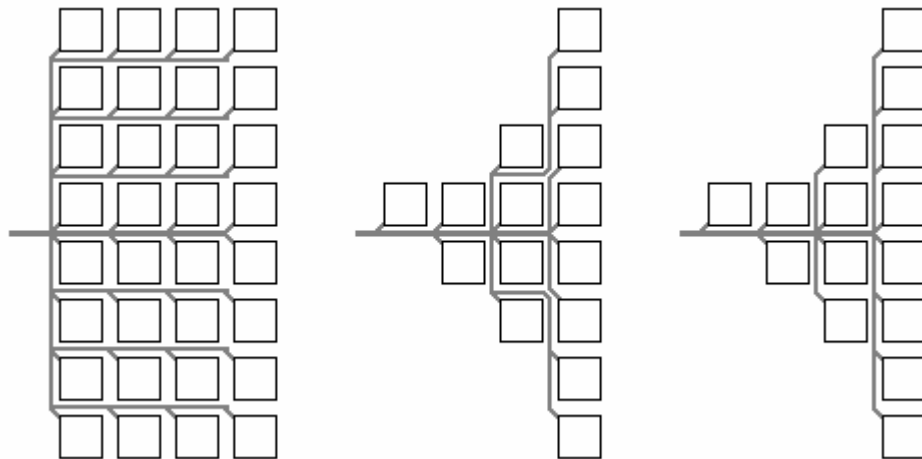


Figura 11: diversi routing nelle due caches.

Il routing usato nelle caches DNUCA (routing 1) impone ad ogni pacchetto di spostarsi sulla giusta riga in corrispondenza della prima colonna. In seguito il pacchetto potrà viaggiare solo in orizzontale senza spostarsi da una riga all'altra. In grigio scuro sono riportati i canali necessari per implementare questo routing.

È possibile creare un equivalente del routing 1 da usare nelle caches TD-NUCA (solo con il simple mapping). Come si vede, ogni pacchetto si sposta prima di riga in riga (verticalmente) e solo dopo prosegue nelle vie successive. A differenza del caso precedente però, si rendono comunque necessari gli spostamenti verticali anche nelle vie successive alla prima.

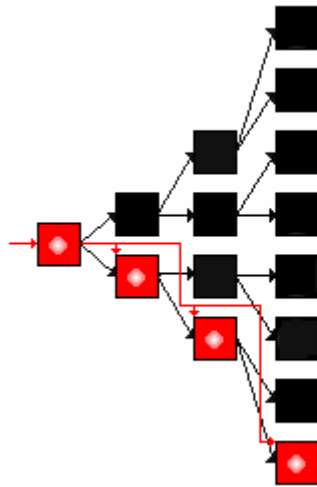
Il routing 2, usato solo nel fair mapping, fa proseguire i pacchetti prima orizzontalmente di colonna in colonna e poi verticalmente fino a giungere alla riga corretta. Questa differenza può cambiare il numero di conflitti generati.

Se invece usassimo il routing 1 con il fair mapping, essendo presenti dei percorsi più lunghi da ogni via alla successiva (le diramazioni dell'albero sono più "lunghe"), non sarebbe possibile ottenere i tempi di routing migliori. Si è allora reso necessario progettare un nuovo routing per ottenere

di nuovo i tempi di figura 4. Il routing 2, che sfrutta i canali mostrati in figura 11c, permette di raggiungere i banchi spostandosi prima sulla giusta colonna e successivamente in modo verticale fino a raggiungere la giusta riga. Così, per raggiungere i banchi appartenenti ad un bank set, sarà lanciato un pacchetto che procederà in orizzontale, e ad ogni colonna si diramerà procedendo sia nella successiva colonna, sia in verticale all'interno della via. Ad ogni istante quindi lo stesso accesso al bank set ha in cache più pacchetti ognuno dei quali procede verticalmente in una diversa via. Questo permette di ottenere, anche per questo mapping, i tempi delle caches rettangolari. Il prezzo che deve pagare questo routing rispetto al precedente è il maggior numero di pacchetti in circolazione nella cache, proprio perché ad ogni colonna il canale centrale crea un nuovo pacchetto per proseguire il cammino verticalmente. Questo maggior numero di pacchetti si produrrà in un maggior numero di conflitti sui canali.

In definitiva le migliori soluzioni adottate per le caches TD-NUCA sono da scegliere fra il simple mapping abbinato all'uso del primo tipo di routing, e il fair mapping abbinato al routing numero 2. La seconda soluzione sarà valutata nei tests di prestazione del capitolo 5.

a) Routing 1 su simple mapping



b) Routing 2 su fair mapping

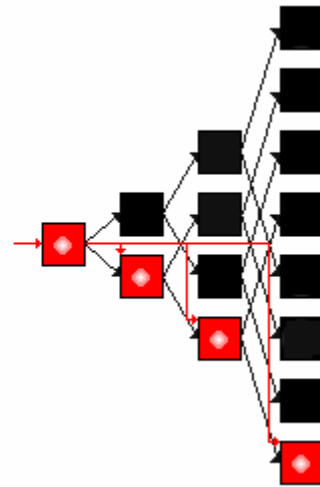


Figura 12: esempio di accesso ad un bank set (in rosso) nei due diversi routing.

I tempi di accesso ai singoli banchi sono gli stessi nei due routing, quello che cambia è il percorso intrapreso dai pacchetti di accesso. Nel routing 2 la lunghezza del percorso preso dai pacchetti (la lunghezza totale della riga rossa) è maggiore rispetto a quella del caso 1 (7 lati di banco contro 6). In conseguenza a questo, le probabilità di conflitti di canale nei due routing possono cambiare, e più precisamente ci aspetteremo più conflitti nel secondo tipo di routing.

In figura 12 è mostrato l'esempio di accesso ad un bank set (costituito dai banchi in rosso) con le due modalità di routing nei rispettivi mapping. La riga in rosso mostra il percorso preso dal pacchetto (o dai pacchetti nel caso di routing 2) nel raggiungere tutti i banchi. Dalla comparazione delle due figure si osserva che l'accesso all'intero bank set (accesso multicast) fa percorrere al pacchetto di accesso 6 canali (si intende 1 canale come il lato di 1 banco) nel caso del routing 1, mentre il pacchetto, o meglio i pacchetti, nel caso del routing 2 ne devono percorrere 7. Questo maggiore impiego di risorse si traduce in un maggior consumo energetico ed in una maggiore probabilità di conflitto fra i pacchetti di accesso. D'altra parte il routing 2 è l'unico modo che si ha con il fair mapping per ottenere i migliori tempi di accesso ai singoli banchi della cache.

3.5 POLITICHE DI RIMPIAZZAMENTO E PROMOZIONE

Fra le politiche previste vi sono quelle che riguardano il rimpiazzamento e la promozione dei blocchi. Il rimpiazzamento si compone nella doppia scelta delle posizioni in cui inserire il nuovo blocco e di quella da cui estrarre il blocco da rimpiazzare. Per l'inserzione sono state considerate le possibilità di inserire nell'ultima via della cache, oppure di inserire in una via scelta a caso. Più precisamente l'inserzione avviene in un blocco vuoto se esiste, oppure seguendo una delle due politiche appena menzionate. Dopo l'inserzione deve essere scelta la posizione dalla quale sarà estratto il blocco da rimpiazzare. L'unica politica scelta è quella di estrazione dal fondo. Ovviamente, se la posizione in cui inserire non coincide con quella da cui estrarre si rende necessaria una operazione di one-copy con la quale viene portato il blocco occupante la posizione di inserzione, nel blocco con la posizione di estrazione (ma solo dopo aver fatto il writeback di quest'ultimo blocco). Per ottimizzare, l'estrazione viene posizionata in un blocco vuoto se esiste, in modo da evitare il writeback, o altrimenti nell'ultima via e solo in questo caso viene effettuato il suo salvataggio in memoria. È da tenere in considerazione che il bank set in cui scegliere la posizione da inserire è quello relativo all'indirizzo del blocco che deve essere inserito in cache, mentre il bank set in cui scegliere la posizione di estrazione è quello riferito all'indirizzo del blocco in cui avviene l'inserzione.

Per quanto riguarda la promozione, questa avviene scambiando il blocco che ha subito la hit con il blocco vittima scelto risalendo le frecce nell'albero del mapping, ed è stato scelto di poter variare in modo flessibile il numero di vie di avanzamento del blocco, da un minimo di 0 (assenza di promozione) ad un massimo corrispondente all'associatività della cache (promozione assoluta). Nella sezione di valutazione delle prestazioni saranno valutate alcune di queste politiche di promozione. Rimane da osservare che è

possibile (anzi è probabile) il caso in cui un blocco debba essere promosso in una delle vie precedenti, e lo scambio con il blocco che occupa la nuova posizione non sia possibile. Il blocco vittima della promozione potrebbe cioè avere un indirizzo incompatibile con il banco in cui è caricato il blocco da promuovere. Per meglio dire, in tale caso il blocco da promuovere proviene da una particolare diramazione dell'albero del mapping (una delle due frecce in entrata possibili nell'andare da una via alla precedente) mentre il blocco vittima può seguire solo l'altra diramazione (nello spostamento verso la via successiva). Le soluzioni proposte in tali casi sono di evitare la promozione, oppure di salvare in memoria ed invalidare il blocco vittima e successivamente eseguire l'operazione di promozione (lo scambio è reso possibile).

3.6 TECNICHE DI RICERCA

Analogamente alle caches rettangolari sono proposte le tecniche di ricerca incrementale e multicast. La ricerca incrementale procede all'accesso del banco di una particolare via solo se il banco relativo alla via precedente ha risposto con una miss. Per contro la ricerca multicast procede all'accesso contemporaneo dei banchi di tutte le vie senza aspettare da queste nessuna risposta. La ricerca multicast ovviamente produrrà più accessi ai banchi, quindi consumerà una quantità maggiore di energia, ma consentirà di avere dei tempi di risposta minori, soprattutto nei casi di hit alle ultime vie o nei casi di miss. Come compromesso a queste tecniche sono nate soluzioni ibride che consentono ricerche parzialmente multicast e parzialmente incrementali.

Una osservazione sulle prestazioni di queste tecniche riguarda l'uso della ricerca incrementale con il routing numero 2. Con il routing 2 infatti si nota come possa essere estremamente inefficiente tale ricerca in quanto un pacchetto dopo aver acceduto al banco di una via, deve tornare nella riga centrale della cache, spostarsi nella via successiva, e risalire di nuovo la colonna per riportarsi sulla giusta riga. I tempi di arrivo ai banchi vengono così allungati notevolmente. Con il routing 1 questo fenomeno non accade poiché i pacchetti per portarsi nella via successiva non devono tornare nella riga centrale ma possono proseguire in orizzontale.

3.7 TECNICA PER LA RIDUZIONE DELLE PERDITE DELL'ENERGIA STATICA

La ricerca [5] applica la tecnica del Dual- V_t CMOS per creare vie alimentate a tensioni differenti. La tecnica Dual- V_t consiste nel calibrare con due (o più) valori la tensione di soglia V_t dei circuiti CMOS in modo differente in ogni sezione del circuito così da compensare la riduzione della velocità di risposta dello stesso con la riduzione delle perdite di energia statica. Questo ci dà lo spunto per applicare tale tecnologia anche ai nostri scopi, creando cioè vie più veloci, le prime, e vie più lente, le ultime. Avremo quindi le ultime vie che oltre ad essere raggiunte dai messaggi in un tempo maggiore rispetto alle precedenti, rispondono esse stesse agli accessi con ritardi maggiori. Notare che il maggiore ritardo introdotto dai tempi di propagazione dei segnali (tempi di routing) non è voluto ma proviene dal problema del wire-delay, mentre il maggiore ritardo di risposta di ogni banco è una tecnica introdotta per permettere un risparmio energetico. In pratica tutto ciò rende attuabile il progetto delle caches NUCA anche per i circuiti attuali nei quali il problema del wire-delay non è ancora determinante sulle prestazioni.

Sempre dall'articolo [5] si può apprendere come l'aumento della tensione di soglia, da 0.2V a 0.4V, possa far scendere il consumo di energia statica di un circuito CMOS alimentato a 0.75V, da 1456nW a 19nW (valori a meno di una costante moltiplicativa) con un aumento dei tempi di propagazione di circa l'88%.

Nel capitolo 5 sarà valutata una configurazione di cache in cui ogni via è costituita da banchi con tempi di risposta calibrati in modo crescente. La prima via ha quindi tempi di risposta di pochi cicli di clock, mentre avanzando fra le vie questo tempo viene dilatato.

3.8 TD-NUCA DECRESCENTI

In ultimo, presentiamo l'idea di rovesciare il triangolo formato dai banchi di cache, in modo da porre la via costituita da più banchi vicina al controller, e lontano da questo la via più piccola. In pratica, il taglio da effettuare sulla matrice di banchi di una cache rettangolare viene effettuato rovesciato. Quello che ci aspettiamo da una cache con questa forma è ovviamente un aumento delle prestazioni rispetto a quella formata dal triangolo crescente in quanto le vie più vicine al controller sono adesso costituite da un maggiore numero di banchi. Le vie più veloci quindi saranno anche le più grandi. Ma, contrariamente a quanto si potrebbe pensare, una cache di questo tipo non consuma più energia rispetto a quella di forma crescente. Infatti quello che cambia fra i due tipi di cache è che quella decrescente ha semplicemente parte dei banchi di memoria avvicinati al controller. Quello che ci aspetteremo sarà semplicemente una riduzione dei tempi di accesso alla cache senza nessuna sostanziale modifica di ogni altro indice. Ovviamente, con riferimento alla sezione precedente, se venisse utilizzata la tecnica della riduzione del consumo di energia impostando i tempi di risposta dei banchi di ogni via in modo crescente, il triangolo crescente consentirebbe un maggiore risparmio di energia rispetto a quello decrescente (i banchi lenti sono in maggior numero) ma fornirebbe prestazioni ancora più basse.

La figura 13 mostra come vengono estese le due modalità di mapping viste nel paragrafo 2 alle caches triangolari decrescenti. Si nota che gli alberi sono semplicemente rovesciati, e che le possibilità di mapping di un blocco nel passare nella via successiva a quella a cui appartiene invece di sdoppiarsi, si riuniscono. Questo fatto causa meno problemi nella promozione, che in questo caso è sempre possibile. Infatti la posizione del blocco da promuovere non risulta mai incompatibile con il blocco vittima di

questa operazione perché per quest'ultimo non esiste la doppia possibilità (doppia freccia) nello spostamento verso la via successiva.

Per il resto continuano a valere immutate le rimanenti politiche di gestione come il mapping verticale, il routing, le tecniche di rimpiazzamento, e le tecniche di ricerca viste nei paragrafi precedenti.

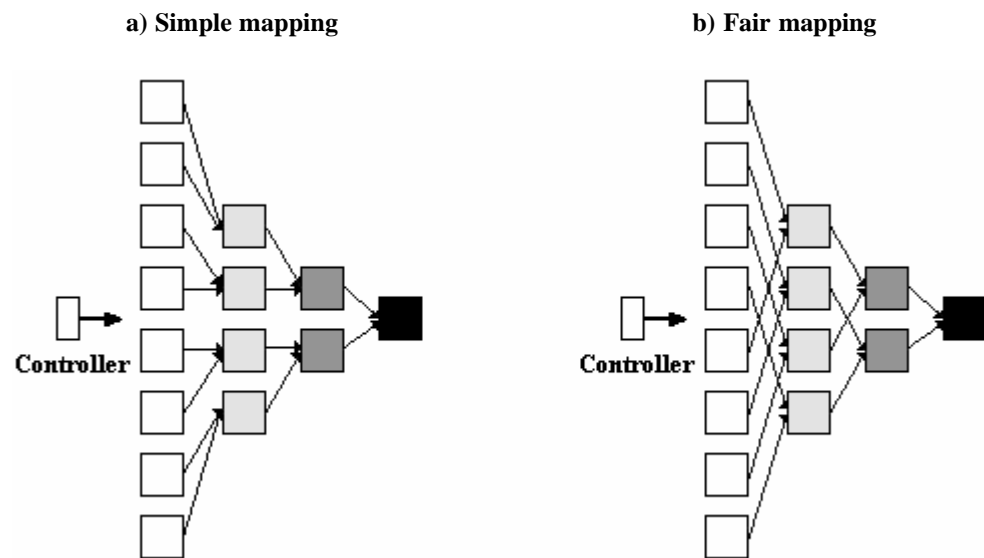


Figura 13: simple mapping e fair mapping per una TD-NUCA decrescente.

Con le caches TD-NUCA decrescenti l'albero del mapping viene rovesciato. Le frecce adesso, invece di sdoppiarsi, si riuniscono. Questo fatto causa meno problemi nella politica di promozione che ora è sempre possibile.

Per il resto, le rimanenti politiche viste nei paragrafi precedenti rimangono immutate.

*Capitolo 4***GLI STRUMENTI USATI****4.1 IL SIMULATORE: D-NUCA SIM-ALPHA 1.0**

In tutto il presente lavoro ogni risultato è stato misurato attraverso una piattaforma di simulazione basata sul diffuso simulatore di architetture Simple Scalar 3.0 [10]. Questo simulatore è un execution-driven ed è basato su una coda di eventi per la comunicazione fra i moduli. In particolare, la versione che abbiamo scelto, la Sim-Alpha 1.0, è basata sull'architettura Alpha 21264 ed è stata validata rispetto a questa da un recente lavoro svolto presso l'Università del Texas di Austin [1]. In più, la versione del simulatore che ci ha permesso di effettuare la presente ricerca ha il supporto alle caches D-NUCA ed è stata messa a punto in una ulteriore ricerca svolta sempre presso l'Università del Texas di Austin ([2] e [6]). I lavori [2] e [6], come descritto in precedenza, mostrano la nuova architettura di caches D-NUCA ed appoggiandosi al simulatore appena citato ne misurano le prestazioni.

In questo lavoro mi sono posto l'obiettivo di migliorare l'esistente progetto delle caches D-NUCA definendo la loro triangolarizzazione ed altre politiche di gestione. Modificando opportunamente il simulatore suddetto ne mostrerò il comportamento nei termini dei seguenti indici: IPC, miss rate, numero di accessi alla L2, numero di accessi ai banchi, numero di conflitti verificatisi. È chiaro che prima di portare alla luce i risultati ottenuti dal mio lavoro di progettazione, dovrò validare il simulatore da me costruito, il TD-NUCA Sim-Alpha, rispetto a quello che per me era il punto di partenza (D-NUCA Sim-Alpha). Nei tests di validazione da me svolti ovviamente mi

porrò nelle stesse condizioni di simulazione del lavoro [2], ed in particolare la configurazione della gerarchia delle caches usate sarà la stessa.

4.2 I BENCHMARKS: SPECINT 2000

Oltre a dover decidere la piattaforma di simulazione si sono dovuti trovare anche i giusti benchmarks su cui basare le prove. Le applicazioni di test scelte appartengono alla collaudata suite SpecInt 2000 [20] ed in particolare ho usato i seguenti: *176.gcc*, *181.mcf*, *256.bzip2*, *300.twolf*. Questi 4 benchmarks sono stati scelti, selezionandoli dall'intero gruppo di Spec, a causa della loro tendenza a generare più miss in L1 così da stressare maggiormente la cache di livello successivo sulla quale è basato il presente lavoro. Inoltre, per massimizzare ulteriormente gli accessi alla cache L2, sono state svolte le prove di simulazione degli Spec ognuna con un proprio FFWD (fast forwarding, ovvero avanzamento rapido della simulazione), in modo da saltare le prime istruzioni che causano pochi accessi in memoria e simulandone solo un certo numero RUN calcolato attraverso ripetute prove degli Spec stessi [2]. La seguente tabella riporta i numeri di FFWD e RUN per ognuno degli Spec utilizzati, ed in più riporta nella terza colonna il numero degli accessi alla L2 per ogni milione di istruzioni.

SPECINT2000	Phase		L2 load acc/ Million instr.
	FFWD	RUN	
176.gcc	2,367B	300M	25.900
181.mcf	5B	200M	260.620
256.bzip2	744B	1B	9.300
300.twolf	511B	200M	22.500

Tabella 6: FFWD e RUN per i 4 SpecInt 2000 utilizzati.

Il FFWD è l'avanzamento rapido del test. Il RUN è il numero di istruzioni simulate.

Gli Spec sono eseguibili compilati ovviamente per l'architettura Alpha 21264 ed ognuno ha i propri input (input REF). In tutte le prove di simulazione presenti in questo lavoro non sono state usate particolari opzioni se non quelle che riguardano la configurazione dell'architettura spiegate nella sezione seguente.

Nei risultati di simulazione riportati nelle tabelle del seguente capitolo 5, i valori degli indici sono stati calcolati come media aritmetica fra gli indici di ogni Spec di cui sopra. Solo per quanto riguarda l'IPC abbiamo scelto di usare la media armonica ($n / \sum (1/IPC)$) in modo da pesare in ugual misura tutti gli Spec (con la media aritmetica gli Spec con un IPC maggiore avrebbero assunto un peso maggiore nella media).

4.3 CONFIGURAZIONE DI BASE DELL'ARCHITETTURA

La configurazione dell'architettura presa in considerazione nelle seguenti simulazioni prevede le impostazioni standard del simulatore Simple Scalar. In tabella 7 sono riportate le principali configurazioni standard dell'architettura Alpha 21264. Per informazioni più precise si rimanda alla letteratura specializzata [10].

Parametro	Valore
Frequenza CPU	500.000.000 Hz
Dimensione della coda di Instruction Fetch	4 istruzioni
Dimensione della coda di commit	11 istruzioni
Dimensione del Reorder Buffer	80 entrate
Numero di Integer ALU	4
Numero di FP ALU	1
Branch predictor	Standard 21264
Intruction TLB e Data TLB	4KB, 128 vie, blocco a 32B, 1 ciclo di latenza
Instruction L1 Cache	64KB, 2 vie, blocco a 64B, 1 ciclo di latenza
Data L1 Cache	64KB, 2 vie, blocco a 64B, 3 cicli di latenza
Dimensione del MSHR (Miss Status Holder Register)	8 entrate per cache
Bus interno al processore	128bit, 1 ciclo di latenza
Bus della memoria (esterno)	128bit, 4 cicli di latenza
Memoria	SDRAM, 1 ciclo di Ras, 1 ciclo di Cas, Single Data Rate, moltiplicatore a 6

Tabella 7: principali parametri di configurazione dell'architettura Alpha 21264 usata nei tests di prestazione del capitolo 5.

In particolare, la gerarchia delle memorie è composta da un TLB dati e da un TLB istruzioni entrambi a 128 vie e di ampiezza 4KB; è composta da una L1 dati e da una L1 istruzioni ciascuna da 64KB, a 2 vie, con blocchi da 64B, 3 cicli di accesso per la parte dati, e a singolo ciclo di accesso per la parte istruzioni; è composta da un bus verso la memoria da 16B di ampiezza e 4 cicli di latenza; è composta infine da una memoria in tecnologia SDRAM con una latenza di 48 cicli per un singolo trasferimento.

Relativamente alla cache L2 la sua definizione avviene inserendo nel file di configurazione una o più stringhe con il seguente schema:

```
<name>:<nsets>:<bsize>:<subblock>:<asso>:<repl>:<lat>:<trans>:
  <prefetch>:<# resources>:<res code>:<res names>
```

Ad esempio, le stringhe:

```
<L2_row0>:<1024>:<64>:<0>:<4>:<L>:<3>:<pipt>:<0>:<1>:<0>:
  <Memory Bus>
```

```
<L2_row1>:<1024>:<64>:<0>:<4>:<L>:<3>:<pipt>:<0>:<1>:<0>:
  <MemoryBus>
```

```
<L2_row2>:<1024>:<64>:<0>:<4>:<L>:<3>:<pipt>:<0>:<1>:<0>:
  <MemoryBus>
```

```
<L2_row3>:<1024>:<64>:<0>:<4>:<L>:<3>:<pipt>:<0>:<1>:<0>:
  <MemoryBus>
```

definiscono una cache L2 con una grandezza di pagina di 4 banki, ed in cui ogni bank set è definito da ogni singola riga. Nell'esempio si nota che ogni banco contiene 1024 insiemi ed ha un tempo di risposta di 3 cicli di clock, che la grandezza del blocco è 64B, che l'associatività è 4, e che la politica di rimpiazzamento è la LRU.

Infine è possibile impostare le dimensioni della matrice di banki che compone la cache come mostrato dal seguente esempio (in teoria le

dimensioni della matrice possono essere diverse dalla struttura della cache per come è stata definita dai precedenti parametri):

```
-nuca:row 4  
-nuca:col 4
```

Nel capitolo successivo per ogni test di simulazione saranno riportati con esattezza questi parametri.

Le dimensioni delle caches provate sempre in considerazione della roadmap SIA 2001 [3] sono: 16MB con una tecnologia di costruzione a 0,50nm, 8MB con una tecnologia di costruzione a 0,70nm, 4MB con una tecnologia di costruzione a 100nm, e 2MB con una tecnologia di costruzione a 130nm.

4.4 IL SIMULATORE MODIFICATO: TD-NUCA SIM-ALPHA

Il simulatore TD-NUCA Sim-Alpha implementa parte delle tecniche riguardanti le caches TD-NUCA definite nel capitolo precedente ed ognuna di esse è impostabile come è mostrato di seguito.

Innanzitutto la forma della struttura dei banchi è definita dal parametro:

```
-nuca:t_banks_number      1      2      4      8
```

In questo esempio la cache è triangolare crescente con 1, 2, 4 e 8 banchi per ogni via. Notare che per attivare la modalità “triangolare” per le caches L2, ovvero per definire la presente opzione, deve essere disattivata la modalità “matriciale” impostando 1 come numero di righe ed impostando la somma totale del numero di banchi come numero di colonne (in questo esempio 15).

Attraverso i seguenti parametri:

```
-nuca:cachebanks_latencies 3      3      3      3
-nuca:interleaved           true
-nuca:random_insertion     false
-nuca:force_promotion      false
-nuca:prom_banks_n         1
```

sono impostate rispettivamente le latenze dei banchi delle quattro vie (nell’esempio ogni via ha banchi con tempi di risposta 3), la modalità di mapping verticale (interlacciata o meno), la tecnica per inserire i blocchi in modo random, la tecnica per forzare la promozione in caso di incompatibilità fra blocco da promuovere e blocco da retrocedere, ed il numero di vie da saltare in ogni promozione. I valori riportati nei precedenti parametri riguardano la configurazione di base dell’architettura della cache relativamente alle prove eseguite nel capitolo 5.

Rimangono escluse dalle implementazioni le tecniche di ricerca incrementale e multicast, le modalità di mapping orizzontale, ovvero simple mapping e fair mapping, e le due tipologie di routing (le caches D-NUCA funzionano solo con il routing 1, mentre le caches TD-NUCA funzionano solo con il routing 2).

*Capitolo 5***VALUTAZIONE DELLE PRESTAZIONI****5.1 VALIDAZIONE**

Questa prima sezione di tests è stata fatta con lo scopo di validare il simulatore Sim-Alpha con supporto alle TD-NUCA rispetto al diretto predecessore, ovvero il Sim-Alpha con supporto alle più semplici D-NUCA. È necessario cioè adattare il comportamento della simulazione di una cache triangolare rispetto all'equivalente rettangolare. Ovviamente per sperare di ottenere le stesse prestazioni dalle due caches abbiamo dovuto riadattare le caches triangolari alle equivalenti rettangolari impostando ogni loro via con la grandezza massima. Ad esempio, da una cache rettangolare 8x4 da 4MB, è stata creata la caches triangolare crescente [1,2,4,8] da 2MB, ed in seguito per creare una cache adatta agli obiettivi della validazione è stata creata la cache [8,8,8,8] da 4MB. Quest'ultimo tipo di cache, che viene detta in configurazione TD-NUCA (o triangolare) anche se poi ha una geometria rettangolare, ci permette di testare il simulatore nello specifico delle modifiche da noi inserite. Nel seguito quindi compareremo il simulatore TD-NUCA Sim-Alpha configurato con una normale cache D-NUCA (viene usato il simulatore modificato ma si deve osservare che le modifiche da noi apportate non influiscono in questo tipo di simulazione), con il simulatore TD-NUCA Sim-Alpha configurato con una cache TD-NUCA rettangolare. In una prima serie di tests abbiamo scelto una configurazione TD-NUCA creata con lo scopo di rendere ininfluenti tutte quelle diversità di comportamento presenti fra i due tipi di caches. A tale fine è stata disattivata la modalità di mapping interlacciato in quanto, come spiegato in

precedenza, le tipologie di interlacciamento usate nei due simulatori risultano diverse. In più, essendo cambiato anche il routing usato dalle due caches, è stata studiata una configurazione adatta in modo da evitare influenze sulle prestazioni da parte dei pacchetti di accesso ai banchi. Proprio per questo abbiamo osservato che il routing 2 (vedi paragrafo 3.4) genera un numero maggiore di pacchetti di accesso rispetto al routing 1, ma garantisce gli stessi tempi di arrivo per ogni banco. Questa differenza quindi si produce solamente in un diverso carico ovvero in un diverso numero di conflitti di canale, ed è per questo che abbiamo forzatamente tolto la possibilità che i pacchetti entrino in conflitto fra loro. Questa ipotesi è ovviamente una condizione di lavoro e non è certamente realistica.

Una volta osservato l'adeguamento delle prestazioni del nostro simulatore configurato con una cache TD-NUCA a quelle relative allo stesso configurato con una cache D-NUCA, abbiamo voluto valutare quantitativamente le differenze appena menzionate. Questo è stato fatto nella successiva serie di tests nei quali abbiamo comparato le due configurazioni di caches rettangolari in presenza della modalità interlacciata e della giusta rilevazione dei conflitti.

La configurazione dell'architettura usata nelle seguenti prove è quella riportata nella sezione 4.3. Gli Spec usati sono quelli di tabella 6 ed i risultati esposti nelle tabelle sono le medie aritmetiche dei risultati di ogni singolo Spec (a parte l'IPC che è calcolato come media armonica).

5.1.1 Modalità non-interlacciata ed a conflitti disabilitati

Nelle seguenti tabelle sono riportati i risultati di validazione relativi alla prima serie di tests. In queste prove è stata attivata l'opzione di modalità non-interlacciata in modo da mappare i blocchi nei banchi con lo stesso criterio in entrambe le configurazioni, ed è stata creata una situazione non reale secondo la quale i pacchetti di accesso alla cache non possono entrare in conflitto di canale fra loro. Con quest'ultima ipotesi gli accessi ai banchi e l'uso dei canali di collegamento nei due tipi di caches avvengono esattamente con gli stessi tempi anche in presenza di carico.

Le caches provate sono da 16MB, 8MB, 4MB, e 2MB. Gli indici posti nella colonna di sinistra riguardano le caches rettangolari D-NUCA valutate con il simulatore modificato. Gli indici posti nella colonna di destra sono relativi alle valutazioni delle caches a geometria rettangolare ma in configurazione TD-NUCA e sono ovviamente misurate con il simulatore modificato.

Nelle tabella 8, come in tutte le seguenti, è possibile leggere dall'alto verso il basso: il numero identificativo del test di simulazione; la configurazione della L2 (D-NUCA o TD-NUCA); la sua geometria, espressa come struttura dei banchi, capacità degli stessi (in numero di set contenuti), e grandezza totale; IPC e miss rate assoluto rilevati dal simulatore; il numero degli accessi alla L2 (che sono sostanzialmente dati dal numero di miss in L1); il numero degli accessi ai banchi di cache (questo valore dà una indicazione del consumo energetico della cache); infine il numero dei conflitti rilevati dai pacchetti di accesso ai banchi.

Come detto in precedenza i valori riportati in tabella sono calcolati come media aritmetica fra i risultati di ogni singolo Spec (a parte l'IPC).

N. test	1	2
Config. L2	D-NUCA	TD-NUCA
Geom. L2	16x16, 1024, 16MB	[16...16], 1024, 16MB
IPC	0,8264	0,8264
Miss rate	0,0076	0,0076
Accessi L2	19.825.159	19.825.233
Accessi ai banchi	427.112.036	427.113.127
Conflitti	0	0

Tabella 8: comparazione di una cache rettangolare da 16MB in modalità non-interlacciata e con la disabilitazione dei conflitti di canale nelle due configurazioni: D-NUCA 16x16 (test n. 1) e TD-NUCA [16,16,16,16,16,16,16,16,16,16,16,16,16,16,16] (test n. 2).

Nel test n. 2 viene utilizzata una cache in configurazione triangolare ma in cui ogni via ha la stessa dimensione creando così una cache rettangolare equivalente a quella del test n. 1. Si può notare che tutti gli indici dei due tests sono esattamente uguali. Da ciò si può evincere che il simulatore modificato ha esattamente gli stessi comportamenti del predecessore se escludiamo le due opzioni disabilite in questa serie di tests.

In particolare nella precedente tabella sono riportati i risultati della comparazione di una cache rettangolare da 16MB in configurazione D-NUCA 16x16 (test n. 1) con una della stessa geometria ma in configurazione TD-NUCA (test n. 2).

Dai risultati si può vedere come tutti gli indici, nella fattispecie IPC, miss rate, numero di accessi alla L2, numero di accessi ai banchi e numero di conflitti (che ovviamente sono 0), siano esattamente uguali in entrambi i tests. Questo dimostra che, almeno sotto le condizioni nelle quali ci siamo posti, la simulazione delle caches TD-NUCA provoca gli stessi comportamenti e fornisce la stessa tempistica rispetto alle corrispondenti D-NUCA.

N. test	3	4
Config. L2	D-NUCA	TD-NUCA
Geom. L2	16x8, 1024, 8MB	[16...16], 1024, 8MB
IPC	0,8084	0,8084
Miss rate (Harm.M)	0,0119	0,0119
Accessi L2	19.830.013	19.830.051
Accessi ai banchi	221.117.509	221.117.890
Conflitti	0	0

Tabella 9: comparazione di una cache rettangolare da 8MB in modalità non-interlacciata e con la disabilitazione dei conflitti di canale nelle due configurazioni: D-NUCA 16x8 (test n. 3) e TD-NUCA [16,16,16,16,16,16,16,16] (test n. 4).

Si nota che in entrambi i tests gli indici hanno gli stessi valori. Anche questi risultati dimostrano l'adeguamento delle prestazioni del simulatore modificato rispetto al diretto originale.

La tabella 9 riporta la stessa comparazione fra caches TD-NUCA e D-NUCA ma per una grandezza di 8MB e dimostra, vista la corrispondenza degli indici, le loro analoghe influenze sulle prestazioni.

N. test	5	6
Config. L2	D-NUCA	TD-NUCA
Geom. L2	8x4, 2048, 4MB	[8...8], 2048, 4MB
IPC	0,7026	0,7026
Miss rate	0,0619	0,0619
Accessi L2	19.829.130	19.829.127
Accessi ai banchi	115.496.830	115.496.775
Conflitti	0	0

Tabella 10: comparazione di una cache rettangolare da 4MB in modalità non-interlacciata e con la disabilitazione dei conflitti di canale nelle due configurazioni: D-NUCA 8x4 (test n. 5) e TD-NUCA [8,8,8,8] (test n. 6).

Valgono le stesse considerazioni fatte per i tests di tabella 8.

N. test	7	8
Config. L2	D-NUCA	TD-NUCA
Geom. L2	4x4, 2048, 2MB	[4...4], 2048, 2MB
IPC	0,6165	0,6165
Miss rate	0,1278	0,1278
Accessi L2	19.839.747	19.839.933
Accessi ai banchi	123.292.960	123.292.720
Conflitti	0	0

Tabella 11: comparazione di una cache rettangolare da 2MB in modalità non-interlacciata e con la disabilitazione dei conflitti di canale nelle due configurazioni: D-NUCA 4x4 (test n. 7) e TD-NUCA [4,4,4,4] (test n. 8).

Valgono le stesse considerazioni fatte per i tests di tabella 8.

Anche per le cache rettangolari da 4MB e 2MB, i cui risultati di simulazione sono riportati nelle precedenti tabelle 10 e 11, si sono riscontrati gli stessi comportamenti per le configurazioni TD-NUCA e D-NUCA. Possiamo

quindi concludere che il simulatore con supporto alle caches triangolari si adatta perfettamente al simulatore con il supporto alle normali caches rettangolari. Queste prove sono anche conferma del fatto che l'unica cosa a cambiare le prestazioni delle due configurazioni di caches sono solamente la modalità di mapping verticale e la tipologia di routing.

5.1.2 Modalità interlacciata

In questa seconda serie di tests di validazione ci proponiamo di valutare l'incidenza delle differenze di gestione dei due tipi di caches D-NUCA e TD-NUCA ovviamente con la stessa geometria rettangolare. Abbiamo quindi usato la configurazione sfruttata nei tests di valutazione di base (vedi paragrafo 5.2). In particolare è stata abilitata la modalità interlacciata ed è stata abilitata la rilevazione dei conflitti di canale creando così una situazione di simulazione reale dell'architettura progettata. Per una maggiore precisione sulla configurazione di base dell'architettura si rimanda al paragrafo successivo. Questi tests ci hanno inoltre permesso di configurare correttamente la restante architettura Alpha 21264 in modo da ottenere risultati di simulazione analoghi agli equivalenti riportati nel lavoro [2].

La caches provate nelle seguenti simulazioni sono le solite rettangolari da 16MB, 8MB, 4MB e 2MB strutturate sempre secondo la roadmap SIA del 2001 [3].

N. test	11	12
Config. L2	D-NUCA	TD-NUCA
Geom. L2	16x16, 1024, 16MB	[16...16], 1024, 16MB
IPC	0,8226	0,8307
Miss rate	0,0076	0,0076
Accessi L2	19.833.786	19.833.821
Accessi ai banchi	427.259.271	427.394.099
Conflitti	21.845.517	62.226.209

Tabella 12: comparazione di una cache rettangolare da 16MB in modalità interlacciata nelle due configurazioni: D-NUCA 16x16 (test n. 11) e TD-NUCA [16,16,16,16,16,16,16,16,16,16,16,16] (test n. 12).

Nel test n. 12 viene utilizzata una cache in configurazione triangolare ma a geometria rettangolare. Si nota che gli indici dei due test differiscono fra loro di pochi punti su 1000, da questo si può dire che il comportamento delle due cache è sostanzialmente lo stesso. La differenza principale tra queste due configurazioni risiede nell'aumentato numero di conflitti fra pacchetti di accesso che comunque non pregiudica le prestazioni generali.

Dai risultati in tabella 12, relativi alla simulazione di una cache da 16MB, si nota che IPC, miss rate (per entrambe le medie), numero di accessi alla L2 e numero di accessi ai banchi sono sostanzialmente uguali fra i due tests. Si può quindi affermare che il comportamento generato dai due diversi tipi di caches è fondamentalmente lo stesso.

L'unico indice a differire fra le due prove è il numero dei conflitti rilevati che risulta molto aumentato nelle caches TD-NUCA a causa della diversa tipologia di routing usata. Le prestazioni comunque non risultano alterate come si vede dall'IPC. Quello che possiamo dedurre da questo aumento nel numero dei conflitti è un possibile aumento nei consumi energetici dato il maggiore numero di pacchetti di accesso in circolazione. Si deve però tenere in considerazione che gran parte del consumo della cache proviene dagli accessi ai banchi (rimasti costanti) e non dal numero di pacchetti generati (che in pratica sono segnali).

N. test	13	14
Config. L2	D-NUCA	TD-NUCA
Geom. L2	16x8, 1024, 8MB	[16...16], 1024, 8MB
IPC	0,8053	0,8130
Miss rate	0,0119	0,0119
Accessi L2	19.838.301	19.838.683
Accessi ai banchi	221.189.368	221.327.275
Conflitti	21.529.324	61.501.912

Tabella 13: comparazione di una cache rettangolare da 8MB in modalità interlacciata nelle due configurazioni: D-NUCA 16x8 (test n. 13) e TD-NUCA [16,16,16,16,16,16,16, 16] (test n. 14).

Valgono le stesse considerazioni fatte per i tests di tabella 12.

Anche la comparazione delle caches da 8MB produce indici sostanzialmente invariati se escludiamo il numero di conflitti rilevati. Le considerazioni fatte per i tests precedenti risultano confermate.

N. test	15	16
Config. L2	D-NUCA	TD-NUCA
Geom. L2	8x4, 2048, 4MB	[8...8], 2048, 4MB
IPC	0,7000	0,7020
Miss rate	0,0622	0,0621
Accessi L2	19.836.712	19.837.208
Accessi ai banchi	115.523.407	115.886.694
Conflitti	22.072.496	37.958.641

Tabella 14: comparazione di una cache rettangolare da 4MB in modalità interlacciata nelle due configurazioni: D-NUCA 8x4 (test n. 15) e TD-NUCA [8,8,8] (test n. 16).

Valgono le stesse considerazioni fatte per i tests di tabella 12.

N. test	17	18
Config. L2	D-NUCA	TD-NUCA
Geom. L2	4x4, 2048, 2MB	[4...4], 2048, 2MB
IPC	0,6100	0,6109
Miss rate	0,1271	0,1269
Accessi L2	19.840.982	19.842.818
Accessi ai banchi	123.339.410	124.123.950
Conflitti	38.589.148	41.671.414

Tabella 15: comparazione di una cache rettangolare da 2MB in modalità interlacciata nelle due configurazioni: D-NUCA 4x4 (test n. 17) e TD-NUCA [4,4,4] (test n. 18).

Valgono le stesse considerazioni fatte per i tests di tabella 12.

Anche per le caches da 4MB e 2MB, i cui risultati di simulazione sono riportati nelle tabelle 14 e 15 rispettivamente, si riscontrano gli stessi risultati. A differenza però delle due prove precedenti si nota in queste che il

numero dei conflitti nelle TD-NUCA rispetto alle equivalenti D-NUCA è aumentato in minore misura. Questo si può spiegare osservando che per le caches di piccole dimensioni il maggior numero di pacchetti generati dal routing 2 non è così rilevante come avveniva per le caches maggiori. Infatti ogni pacchetto generato dal routing 2 deve percorrere la cache nella sua larghezza, ma se questa è piccola, la probabilità di generare conflitti non viene aumentata molto.

In conclusione di questa fase di validazione possiamo ritenere nel seguito delle simulazioni l'effettiva validazione delle prestazioni delle caches TD-NUCA rispetto a quelle delle caches D-NUCA.

5.2 CONFIGURAZIONE DI BASE

In questa sezione valuteremo le prestazioni delle caches TD-NUCA a geometria triangolare. L'obiettivo di questa serie di prove è di riuscire a trovare una configurazione che permetta un risparmio sulle dimensioni fisiche della cache (il 50% circa) con una minima perdita sulle prestazioni. Le prove oggetto di questo paragrafo comprenderanno caches in configurazioni triangolari crescenti e triangolari decrescenti le quali saranno messe a confronto con le rettangolari equivalenti valutate nel paragrafo precedente.

La configurazione di base delle caches valutate qui di seguito comprenderà una latenza alla parte dati dei bank di 3 cicli di clock e una latenza per l'accesso alla parte tag dei bank di 2 cicli. La politica usata per la gestione del rimpiazzamento dei blocchi è la LRU, gli indirizzi considerati nella L2 sono indirizzi fisici, non si fa uso del subblocking, e non è presente la L3. Per quanto riguarda le politiche tipicamente TD-NUCA è usato il fair mapping, la modalità interlacciata, la promozione di 1 via ad ogni hit, non è usato il random insertion e non è usata la force promotion (vedi 4.4).

Le caches provate sono le solite viste nella sezione precedente, ovvero da 16MB, 8MB, 4MB e 2MB, con l'aggiunta di una ulteriore configurazione da 1MB. Fra gli indici riportati nelle successive tabelle si trova anche un indicatore sulla via media nella quale si sono avute le hit nelle prove. Questo indice è calcolato come media aritmetica fra le vie, ognuna pesata con il numero di hit rilevate e con la dimensione della via stessa, secondo la formula $(\text{via} * \text{n_hit} * \text{dimen_via}) / (\text{n_hit} * \text{dimen_via})$.

Gli Spec usati sono sempre quelli descritti nel paragrafo 4.2, ognuno con le relative opzioni. Analogamente, la configurazione dell'architettura rimane quella descritta nel paragrafo 4.3.

N. test	12	21	22
Config. L2	TD-NUCA	TD-NUCA	TD-NUCA
Geom. L2	[16...16], 1024, 16MB	[2,2,2,2,4,4,4,4,8,8,8,8,16,16,16,16], 1024, ~8MB	[16,16,16,16,8,8,8,8,4,4,4,4,2,2,2,2], 1024, ~8MB
IPC	0,8307	0,7504	0,8268
Miss rate	0,0076	0,0136	0,0100
Accessi ai banchi	427.394.099	437.241.098	427.320.337
Via media con hit	1,68	7,23	1,58

Tabella 16: comparazione di una cache rettangolare da 16MB con le equivalenti triangolari crescente e decrescente da circa 8MB.

Si nota come l'IPC del test 22 sia rimasto costante. Per il test 21 invece l'IPC risulta ridotto dell'8% (0,7504). Il miss rate molto basso (0,76%) ci indica che il working set è più piccolo di 16MB e quindi che le miss sono gran parte di cold. La diminuzione delle prestazioni del test 21 rispetto a quelle del test 22 non è quindi da ricercare nel numero di miss che è rimasto invariato, ma bensì nella posizione degli hit. Come si vede dall'ultima riga della tabella, le caches crescenti in media causano hit fra la settima e la ottava via (7,23), mentre per gli altri due test le hit hanno luogo appena dopo la prima via (circa 1,60). Il più basso IPC delle caches crescenti si spiega perciò con una maggiore lontananza dei blocchi che fanno hit. La caches decrescente invece risulta più contenuta intorno al controller e proprio per questo fornisce prestazioni migliori.

Si nota infine come le caches crescenti consumino più rispetto alle decrescenti (437M di accessi ai banchi contro 427M). Anche questo è da imputare ad una maggiore lontananza degli accessi con hit avuti in cache per il test 21.

In queste prime prove è stata comparata una cache rettangolare da 16MB con le equivalenti triangolari da circa 8MB. Si può notare subito un miss rate per il test 12 estremamente basso: ciò significa che il working set dei benchmarks provati è minore di 16MB, e si può trovare conferma a questo osservando anche che le vie finali (le più lontane dal controller) delle caches qui provate, risultano vuote e mai accedute con successo. Le miss sono quindi solo di cold.

Osservando l'IPC si nota come le caches crescenti forniscano prestazioni ridotte rispetto agli altri due tests (0,7504 contro 0,8307 e 0,8268). Per

spiegare questo comportamento abbiamo calcolato per le presenti prove, la via media nella quale si hanno le hit. La struttura crescente della cache nel test 21 porta ad una maggiore lontananza dei blocchi dal controller che si conferma con una via media per le hit di 7,23. Ciò vuol dire che in media un accesso in cache deve accedere fra la settima via e la ottava con un notevole tempo di ritardo. Nel caso delle caches rettangolari e triangolari decrescenti, la via media con hit è di circa 1,60. Questo spiega la diminuzione delle prestazioni delle caches crescenti.

Da notare infine che il numero degli accessi ai banchi della L2 crescente è salito a 437M determinando un aumento nel consumo di queste caches. Anche questa tendenza si spiega con la maggiore lontananza dal controller dei blocchi utili che implica il dover accedere ad un maggior numero di vie per ogni accesso alla cache.

N. test	14	23	24
Config. L2	TD-NUCA	TD-NUCA	TD-NUCA
Geom. L2	[16...16], 1024, 8MB	[2,2,4,4,8,8,16,16], 1024, ~4MB	[16,16,8,8,4,4,2,2], 1024, ~4MB
IPC	0,8130	0,6241	0,6782
Miss rate	0,0119	0,0711	0,0573
Accessi ai banchi	221.327.275	228.314.477	218.395.057
Via media con hit	1,65	5,09	1,37

Tabella 17: comparazione di una cache rettangolare da 8MB con le equivalenti triangolari crescente e decrescente da circa 4MB.

Analogamente alla comparazione di tabella 15 abbiamo una inefficienza delle caches crescenti rispetto a quelle decrescenti. In queste prove si iniziano a sentire le prime miss di conflitto (il miss rate è sensibilmente salito per le cache triangolari) che provoca un abbassamento delle prestazioni anche per le cache triangolari decrescenti (il loro IPC vale 0,6782 ed è più basso del 13% rispetto a quello delle caches rettangolari).

Osservando la tabella 17 si può di nuovo notare che l'IPC delle caches triangolari crescenti è nettamente minore sia all'IPC relativo al test della cache rettangolare sia a quello della cache decrescente. Le spiegazione è data ancora una volta da una maggiore lontananza dei blocchi utili in cache (lo si può vedere dalla via media delle hit).

In questa tabella però si nota una diminuzione dell'IPC anche per le caches decrescenti. Questa diminuzione è avvenuta a causa di un innalzamento del miss rate. La minore dimensione della cache triangolare decrescente quindi si inizia a fare sentire con una diminuzione del 13% sull'IPC (da 0,8130 della cache rettangolare da 8MB a 0,6782 della cache triangolare da 4MB). Infine si nota ancora l'aumento del consumo energetico delle caches crescenti, in quanto gli accessi ai banchi sono saliti a 228M.

N. test	16	25	26
Config. L2	TD-NUCA	TD-NUCA	TD-NUCA
Geom. L2	[8...8], 2048, 4MB	[1,2,4,8], 2048, ~2MB	[8,4,2,1], 2048, ~2MB
IPC	0,7020	0,5699	0,6158
Miss rate	0,0621	0,1611	0,1378
Accessi ai banchi	115.886.694	121.536.471	114.759.756
Via media con hit	1,45	3,43	1,18

Tabella 18: comparazione di una cache rettangolare da 4MB con le equivalenti triangolari crescente e decrescente da circa 2MB.

La cache crescente continua ad essere nettamente peggiore, sia per quanto riguarda l'IPC (0,5699) sia per il numero degli accessi ai banchi (121M contro 115M). La cache decrescente perde ancora il 9% rispetto all'IPC delle caches rettangolari.

Per le caches da 4MB si continua a notare la stessa tendenza avuta per la comparazione precedente. Le cache crescenti perdono il 13% sulle

rettangolari (0,5699 contro 0,7020), consumano di più (121M contro 115M) e si nota ora un alto tasso di miss (16,11%). L'aumento del miss rate anche rispetto a quello delle caches decrescenti della stessa dimensione è dovuto ad una maggiore media nelle vie che hanno avuto hit (3,43 contro 1,18). Infatti, in queste caches a 4 vie, un blocco prima di essere rimpiazzato dalla stessa deve essere retrocesso nell'ultima via (si ricorda che i blocchi sono estratti solo dall'ultima via), percorrendo solo 0,57 "vie" per le caches crescenti e ben 2,82 per le caches decrescenti. Si ha quindi una maggiore probabilità che alcuni blocchi ancora utili (frequentemente acceduti secondo la terminologia LRU) vengano rimossi dalla cache crescente poiché devono retrocedere di una quantità minore di vie. Per questo le caches decrescenti contengono mediamente blocchi più importanti (più acceduti) e causano una percentuale di miss molto minore della relativa percentuale delle caches crescenti, anche a parità di dimensione di cache.

Le caches decrescenti perdono comunque il 9% sulle caches rettangolari (0,6158 contro 0,7020) a causa di un innalzamento del miss rate (13,78%). La ridotta dimensione della cache triangolare, come è avvenuto per i risultati della tabella 17, è quindi rilevante.

Per quanto riguarda il consumo energetico, le caches decrescenti causano lo stesso numero di accessi ai banchi mentre le cache crescenti portano il numero degli accessi intorno ai 121M contro i 115M delle caches rettangolari.

N. test	18	27	28
Config. L2	TD-NUCA	TD-NUCA	TD-NUCA
Geom. L2	[4...4], 2048, 2MB	[1,1,2,4], 2048, ~1MB	[4,2,1,1], 2048, ~1MB
IPC	0,6109	0,5077	0,5532
Miss rate	0,1269	0,2802	0,2676
Accessi ai banchi	124.123.950	126.111.285	121.377.767
Via media con hit	1,82	3,44	1,31

Tabella 19: comparazione di una cache rettangolare da 2MB con le equivalenti triangolari crescente e decrescente da circa 1MB.

Le caches crescenti perdono il 10% sulle rettangolari (0,5077 contro 0,6109), le caches decrescenti perdono invece il 6% (0,5532). Anche per le caches decrescenti il miss rate è ora aumentato al 26,76% ma rimane comunque inferiore a quello delle caches crescenti.

L'osservazione della tabella 19 ci conferma le considerazioni fatte per la tabella 18: si nota infatti che il miss rate della cache crescente è molto aumentato pregiudicando le prestazioni di questa simulazione (l'IPC è diminuito del 10%). Nel test 28 si ha comunque un netto aumento del miss rate (indice che il working set è maggiore di 1MB) che causa una diminuzione dell'IPC del 6% anche per le caches decrescenti. La cache decrescente rimane comunque migliore, sia come IPC che come numero di accessi ai banchi (e quindi come risparmio energetico), della cache crescente.

N. test	19	29	30
Config. L2	TD-NUCA	TD-NUCA	TD-NUCA
Geom. L2	[4...4], 1024, 1MB	[1,1,2,4], 1024, ~512KB	[4,2,1,1], 1024, ~512KB
IPC	0,5334	0,4368	0,4656
Miss rate	0,1879	0,4636	0,4851
Accessi ai banchi	72.571.791	71.825.801	69.101.410
Via media con hit	2,14	3,56	1,29

Tabella 20: comparazione di una cache rettangolare da 1MB con le equivalenti triangolari crescente e decrescente da circa 512KB.

Le caches crescenti perdono il 10% sulle rettangolari (0,4368 contro 0,5334), le caches decrescenti perdono invece il 7% (0,4656). Il miss rate delle caches decrescenti è aumentato al 48,51%.

Anche dall'osservazione della tabella 20 si riscontrano le stesse tendenze: i miss rate delle caches triangolari sono molto aumentati (46,36% e 48,51%) e le prestazioni hanno subito la stessa diminuzione percentuale (10% in meno sull'IPC per le caches crescenti e 7% in meno per quelle decrescenti). La cache decrescente si dimostra ancora migliore, sia come IPC che come numero di accessi ai banchi (e quindi come risparmio energetico), della cache crescente.

Una osservazione importante nasce dalla comparazione trasversale del test 28 relativo ad una cache triangolare da 1MB con il test 19 relativo ad una cache rettangolare ancora da 1MB (tabella 21). Per queste due caches della stessa dimensione ma organizzate in modo diverso si riscontra un IPC maggiore del 2% da parte della cache triangolare (0,5532 contro 0,5334 rispettivamente).

Una prima conclusione importante è quindi che una organizzazione triangolare a parità di dimensione totale della cache fornisce prestazioni migliori riuscendo comunque a tenere sotto controllo il consumo energetico. Tutto questo è da imputare ad un aumento della dimensione massima della pagina di memoria che limita le miss di conflitto, ed è da imputare ad una maggiore vicinanza al controller dei blocchi importanti (lo si nota dall'ultima riga della tabella), fatto che riesce a diminuire le latenze di accesso ed il consumo di energia necessario ad ogni ricerca.

N. test	19	28
Config. L2	TD-NUCA	TD-NUCA
Geom. L2	[4...4], 1024, 1MB	[4,2,1,1], 2048, ~1MB
IPC	0,5334	0,5532
Miss rate	0,1879	0,2676
Accessi ai banchi	72.571.791	121.377.767
Via media con hit	2,14	1,31

Tabella 21: comparazione di una cache rettangolare da 1MB con una cache triangolare decrescente da 1MB.

L'IPC della cache triangolare da 1MB risulta maggiore di quello della cache rettangolare da 1MB (0,5532 e 0,5334 rispettivamente), indice di una migliore strutturazione e gestione delle suddette caches triangolari.

Nel grafico 1 sono riportati gli IPC di ogni test valutato in questo paragrafo. È evidente che per un working set molto più piccolo della dimensione delle caches, le prestazioni sono molto simili nei tre tests (primo gruppo di colonne). Andando a valutare working set maggiori (o equivalentemente caches più piccole) si riscontra un forte decadimento delle prestazioni per le

cache triangolari crescenti mentre le cache decrescenti riescono a fornire prestazioni non lontane dalle equivalenti rettangolari. Con la diminuzione della capacità della cache comunque, l'IPC delle caches triangolari decrescenti tende ad avvicinarsi a quello delle caches rettangolari.

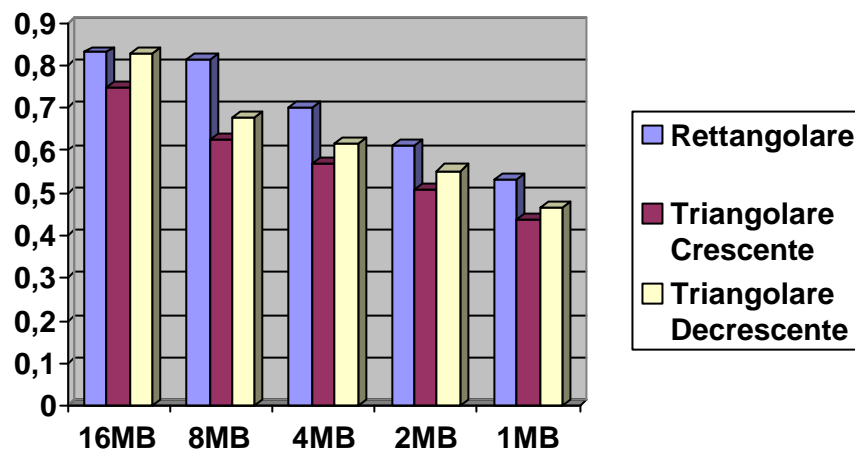


Grafico 1: IPC a confronto.

Un altro fatto importante riguarda la comparazione di caches della stessa dimensione ma con diversa organizzazione. Le prestazioni delle caches da circa 8MB, 2MB e 1MB triangolari decrescenti (colonna gialla del primo, terzo e quarto gruppo di colonne) sono maggiori dell'1%, dell'1% e del 2% rispetto a quelle relative alle caches da 8MB, 2MB e 1MB rettangolari (colonna azzurra del secondo, quarto e quinto gruppo). A parità di dimensione dello spazio di silicio occupato, una cache triangolare è più performante di una cache rettangolare).

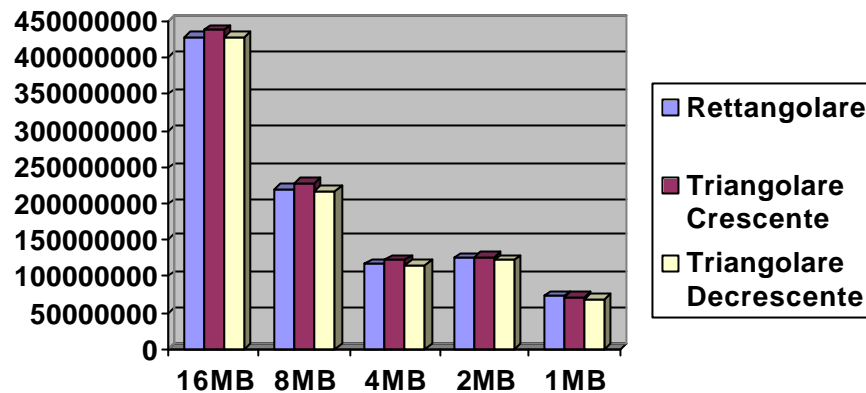


Grafico 2: accessi ai banchi a confronto.

Nel grafico 2 è riportata la quantità di accessi ai banchi di ogni tipo di cache. Questi valori sono proporzionali al consumo energetico della cache. Si nota che caches di grandi dimensioni hanno un numero maggiore di accessi rispetto a quelle più piccole. Inoltre le cache triangolari crescenti hanno sempre un consumo maggiore rispetto alle equivalenti decrescenti, le quali hanno un numero di accessi pressoché identico a quello delle caches rettangolari. In conclusione possiamo dire che le cache decrescenti sono migliori delle corrispondenti crescenti sia sotto l'aspetto delle prestazioni, sia sotto l'aspetto dei consumi energetici.

In più, se volessimo progettare un sistema con particolare riguardo ai consumi, potremmo prevedere una configurazione triangolare per la cache L2 dimezzandone la dimensione e la complessità e peggiorando sensibilmente le prestazioni rispetto alla normale struttura rettangolare. In alternativa se volessimo progettare un sistema in funzione delle sue prestazioni possiamo inserire una configurazione triangolare della stessa dimensione della equivalente rettangolare riuscendo persino ad aumentare le prestazioni come abbiamo visto per le caches da 1MB.

5.3 CONFIGURAZIONE AVANZATA

In questa sezione finale abbiamo voluto valutare l'impatto che alcune particolari politiche hanno sulle caches triangolari. Visti i risultati della sezione precedente abbiamo preso in considerazione solo le caches triangolari decrescenti dimostratesi migliori di quelle crescenti. Per eseguire la valutazione di ogni politica è stata presa la configurazione base della cache TD-NUCA e su di questa è stata attivata di volta in volta l'opzione relativa alla politica voluta.

L'obiettivo di questa sezione è di riuscire a trovare dei miglioramenti nell'organizzazione e nella gestione della cache in modo da ottenere risultati migliori sia dal punto di vista del risparmio che da quello delle prestazioni. La cache provata avrà una grandezza di 1MB.

N. test	18	28	31
Config. L2	TD-NUCA	TD-NUCA	TD-NUCA
Geom. L2	[4...4], 2048, 2MB	[4,2,1,1], 2048, ~1MB	[4,2,1,1], 2048, ~1MB
Opzioni attivate	Nessuna	Nessuna	Latenze dei banchi crescenti
IPC	0,6109	0,5532	0,5037
Miss rate	0,1269	0,2676	0,2698
Accessi ai banchi	124.123.950	121.377.767	121.309.716

Tabella 22: comparazione della configurazione base di caches triangolari da 1MB con alcune configurazioni avanzate della stessa.

Nel test n. 31 viene valutata una configurazione in cui i banchi di ogni via hanno una latenza di accesso decrescente all'aumentare delle vie. Questo permette di realizzare su silicio le vie con differenti tecnologie (vedi Dual- V_t CMOS) permettendo un risparmio energetico. Ovviamente le prestazioni risultano minori di circa il 5% rispetto ad una triangolare decrescente in configurazione base e dell'11% rispetto ad una cache rettangolare.

N. test	32	33	34
Config. L2	TD-NUCA	TD-NUCA	TD-NUCA
Geom. L2	[4,2,1,1], 2048, ~1MB	[4,2,1,1], 2048, ~1MB	[4,2,1,1], 2048, ~1MB
Opzioni attivate	Inserzione casuale	Promozione a 2 vie	Senza promozione
IPC	0,5327	0,5426	0,5352
Miss rate	0,2415	0,2878	0,3710
Accessi ai banchi	131.759.461	121.432.414	112.913.256

Tabella 23: comparazione della configurazione base di caches triangolari da 1MB con alcune configurazioni avanzate della stessa (segue).

Nel test n. 32 è stata provata la politica di inserzione di un nuovo blocco in cache in una via scelta in modo random. Inserendo i blocchi in posizioni più vicine al controller rispetto alla configurazione base, si ha un miss rate minore rispetto a quello del test di base. L'IPC non riesce comunque a beneficiare di questo vantaggio.

Nel test n. 33 abbiamo provato a promuovere il blocco che ha subito la hit di due vie anziché di una come accade nel test di base. Si nota come questo beneficio non sia abbastanza se confrontato con il maggior lavoro che la cache deve effettuare per l'operazione di promozione. L'IPC quindi risulta ridotto dell'1% rispetto alla configurazione base.

Nel test n. 34, ultimo fra le nostre prove, sono descritte le prestazioni della solita cache ma in cui è stata disabilitata la promozione. Si nota subito come l'IPC sia diminuito del 2% circa, e come il numero di accessi ai banchi sia diminuito di circa il 7% in conseguenza ad un minor lavoro che deve effettuare la cache.

Con il test 31 è stata verificata la perdita di prestazioni nel caso i banchi di ogni via avessero un ritardo maggiore rispetto a quello usato nei test di base. Con riferimento a quanto detto nel paragrafo 3.7 ed al lavoro [5] relativamente alle differenti tecnologie implementative delle vie in una cache (Dual- V_t CMOS) abbiamo impostato il ritardo dei banchi di ogni via in modo crescente e precisamente a 3, 4, 5 e 6 cicli di clock rispettivamente. I risultati sono stati un peggioramento delle prestazioni (in termini di IPC) del 11% sulle normali caches rettangolari e del 5% sulle caches triangolari in configurazione base. Il numero di accessi ai banchi sono rimasti

sostanzialmente costanti, ma il consumo energetico è ridotto a livello di ogni singolo accesso ai banchi. Riprendendo infatti i risultati mostrati nel suddetto paragrafo si può stimare che l'accesso ad ogni via successiva alla prima dissipi una minor quantità di energia statica fino all'ultima (la quarta) nella quale con tempi di accesso doppi (si passa da un ritardo di 3 a 6) viene dissipata approssimativamente l'1% di energia rispetto a quella dissipata nella prima. In definitiva per ottenere un considerevole risparmio energetico è possibile adottare questa soluzione riducendo le prestazioni di una piccola percentuale.

Il test 32 valuta l'impatto di una inserzione dei blocchi in una via scelta in modo casuale. Dai risultati esposti in tabella si nota che il miss rate è ridotto del 2% rispetto alla configurazione base (da 26,76% a 24,15%). Questo è dovuto al fatto che in cache permangono blocchi più importanti poiché questi sono inseriti in vie intermedie. L'IPC comunque non riesce a beneficiare di questo vantaggio poiché sulle prestazioni influisce adesso anche la politica one-copy che si rende necessaria alla inserzione casuale (nell'inserzione dal fondo della configurazione base è usata una più leggera zero-copy). La pesantezza della politica one-copy si riscontra anche dal numero degli accessi ai banchi aumentati del 9% (da 121M a 132M).

Il test 33 mostra le performance di una promozione a due vie. In questo test il miss rate è aumentato del 2% e conseguentemente l'IPC è sceso dell'1%. Questa politica in pratica velocizza troppo la dinamicità dei blocchi rischiando di allontanare dal controller blocchi importanti.

Infine il test 34 valuta le cache TD-NUCA in assenza di promozione. Questa politica comporta una diminuzione dell'IPC del 2% ma permette una diminuzione dell'accesso ai banchi del 7% (da 121M a 113M). Risulta quindi efficace allo scopo di ridurre il consumo energetico.

Osservando il grafico riepilogativo qui sotto, si nota come la configurazione base sia quella che riesce a dare le migliori prestazioni in termini di IPC. Se invece considerassimo la minimizzazione dei consumi, è possibile adottare una soluzione con latenze dei banchi crescenti magari insieme ad una politica senza promozione.

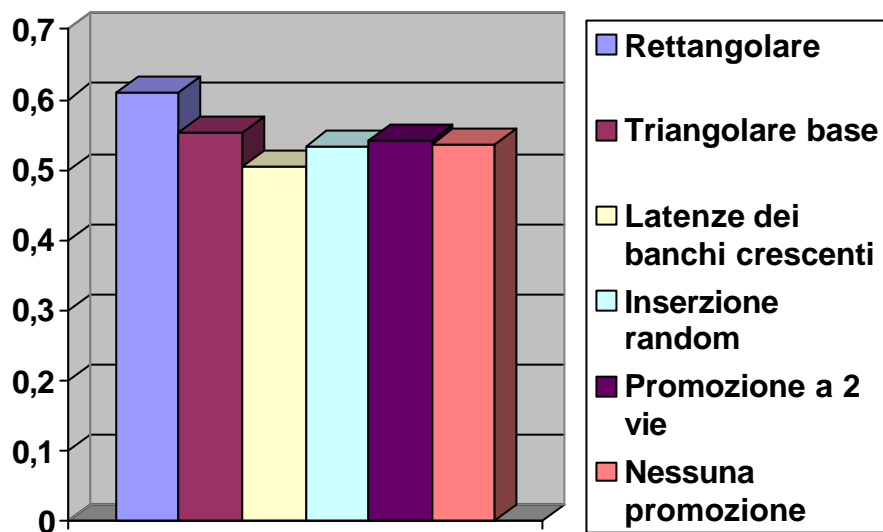


Grafico 3: comparazione degli IPC fra le configurazioni di base ed avanzate (1MB).

*Capitolo 6***CONCLUSIONI**

Nel corso di questo lavoro è stata definita la struttura logica di una cache D-NUCA Triangolare. In particolare sono state scelte due configurazioni: una a geometria crescente, nella quale la grandezza di ogni via è raddoppiata all'allontanarsi del controller, ed una a geometria decrescente nella quale la dimensione delle vie è ogni volta dimezzata. Un primo risultato importante che abbiamo ottenuto riguarda i netti vantaggi che la cache TD-NUCA decrescente offre su una TD-NUCA crescente di pari dimensione. E' risultato infatti che una cache a geometria decrescente si comporta meglio dell'equivalente crescente, sia dal punto di vista energetico che da quello delle prestazioni, essenzialmente a causa delle minori latenze che offre ad ogni accesso.

Abbiamo proseguito gli studi comparando le normali caches D-NUCA rettangolari con le TD-NUCA decrescenti. Comparando caches D-NUCA da 8MB con caches TD-NUCA da 4MB, caches D-NUCA da 4MB con caches TD-NUCA da 2MB, caches D-NUCA da 2MB con caches TD-NUCA da 1MB ed infine caches D-NUCA da 1MB con caches TD-NUCA da 512KB si è verificato un calo di prestazioni del 13%, del 9%, del 6% e del 7% rispettivamente (vedi paragrafo 5.2), ma a fronte di una diminuzione della superficie della cache di circa il 50%. E' stata poi messa a confronto una cache D-NUCA da 1MB con una cache TD-NUCA da 1MB. La cache a configurazione triangolare è risultata più veloce del 2% rispetto a quella rettangolare senza aumentare molto il numero di accessi ai banchi (tabella 21).

Infine sono state messe a punto ed in seguito valutate alcune politiche avanzate. Fra i risultati più importanti è da sottolineare la tecnica Dual- V_t CMOS la quale ci ha permesso di costruire delle vie con tempi di ritardo

crescenti fornendo un notevole risparmio energetico al costo di un 5% in meno sull'IPC (tabella 22). Un'ultima osservazione degna di nota riguarda l'uso della promozione nelle caches studiate. E' stato infatti verificato che in assenza di promozione si verifica una perdita di prestazioni del 2%, ma gli accessi ai banchi di cache vengono ridotti del 7% (tabella 23).

Alcuni studi lasciati aperti possono riguardare la ricerca di una geometria non necessariamente triangolare che consenta una ulteriore ottimizzazione dei risultati.

Appendice A

PRESTAZIONI DEI SINGOLI BENCHMARKS

Sono riportati qui di seguito i risultati di simulazione di ogni singolo Spec provato nei nostri tests. Per maggiori informazioni sugli Spec si rimanda al paragrafo 4.2. Per maggiori informazioni sulla configurazione usata nei tests si rimanda al paragrafo 4.3. Le seguenti tabelle sono organizzate in modo del tutto uguale a quelle del capitolo 5.

A.1 176.gcc

N. test	1	2
Config. L2	D-NUCA	TD-NUCA
Geom. L2	16x16, 1024, 16MB	[16...16], 1024, 16MB
IPC	1,0872	1,0872
Miss rate	0,0019	0,0019
Accessi L2	24.011.668	24.011.668
Accessi ai banchi	532.338.411	532.338.411
Conflitti	0	0

Tabella 24: comparazione di una cache rettangolare da 16MB in modalità non-interlacciata e con la disabilitazione dei conflitti di canale nelle due configurazioni: D-NUCA 16x16 (test n. 1) e TD-NUCA [16,16,16,16,16,16,16,16,16,16,16,16,16,16,16,16] (test n. 2).

Tutti gli indici sono perfettamente corrispondenti.

N. test	3	4
Config. L2	D-NUCA	TD-NUCA
Geom. L2	16x8, 1024, 8MB	[16...16], 1024, 8MB
IPC	1,089	1,089
Miss rate	0,0019	0,0019
Accessi L2	24.011.734	24.011.734
Accessi ai banchi	269.890.911	269.890.911
Conflitti	0	0

Tabella 25: comparazione di una cache rettangolare da 8MB in modalità non-interlacciata e con la disabilitazione dei conflitti di canale nelle due configurazioni: D-NUCA 16x8 (test n. 3) e TD-NUCA [16,16,16,16,16,16,16,16] (test n. 4).

Tutti gli indici sono perfettamente corrispondenti.

N. test	5	6
Config. L2	D-NUCA	TD-NUCA
Geom. L2	8x4, 2048, 4MB	[8...8], 2048, 4MB
IPC	1,2831	1,2831
Miss rate	0,0025	0,0025
Accessi L2	24.011.996	24.011.996
Accessi ai banchi	138.655.092	138.655.092
Conflitti	0	0

Tabella 26: comparazione di una cache rettangolare da 4MB in modalità non-interlacciata e con la disabilitazione dei conflitti di canale nelle due configurazioni: D-NUCA 8x4 (test n. 5) e TD-NUCA [8,8,8,8] (test n. 6).

Tutti gli indici sono perfettamente corrispondenti.

N. test	7	8
Config. L2	D-NUCA	TD-NUCA
Geom. L2	4x4, 2048, 2MB	[4...4], 2048, 2MB
IPC	1,319	1,319
Miss rate	0,0182	0,0182
Accessi L2	24.013.499	24.013.499
Accessi ai banchi	154.216.937	154.216.937
Conflitti	0	0

Tabella 27: comparazione di una cache rettangolare da 2MB in modalità non-interlacciata e con la disabilitazione dei conflitti di canale nelle due configurazioni: D-NUCA 4x4 (test n. 7) e TD-NUCA [4,4,4,4] (test n. 8).

Tutti gli indici sono perfettamente corrispondenti.

Relativamente a 176.gcc si può dire che il comportamento delle caches TD-NUCA è in sintonia con quello delle normali caches D-NUCA.

N. test	11	12
Config. L2	D-NUCA	TD-NUCA
Geom. L2	16x16, 1024, 16MB	[16...16], 1024, 16MB
IPC	1,1314	1,1277
Miss rate	0,0019	0,0019
Accessi L2	24.008.825	24.008.482
Accessi ai banchi	532.291.268	532.443.533
Conflitti	16.725.778	39.346.148

Tabella 28: comparazione di una cache rettangolare da 16MB in modalità interlacciata nelle due configurazioni: D-NUCA 16x16 (test n. 11) e TD-NUCA [16,16,16,16,16,16,16,16,16,16,16,16] (test n. 12).

Gli indici sono sostanzialmente invariati ad eccezione del numero di conflitti che è raddoppiato per la cache TD-NUCA.

N. test	13	14
Config. L2	D-NUCA	TD-NUCA
Geom. L2	16x8, 1024, 8MB	[16...16], 1024, 8MB
IPC	1,1333	1,1297
Miss rate	0,0019	0,0019
Accessi L2	24.008.627	24.008.396
Accessi ai banchi	269.860.915	270.013.252
Conflitti	16.703.150	39.366.005

Tabella 29: comparazione di una cache rettangolare da 8MB in modalità interlacciata nelle due configurazioni: D-NUCA 16x8 (test n. 13) e TD-NUCA [16,16,16,16,16,16,16,16] (test n. 14).

Gli indici sono sostanzialmente invariati ad eccezione del numero di conflitti che è raddoppiato per la cache TD-NUCA.

N. test	15	16
Config. L2	D-NUCA	TD-NUCA
Geom. L2	8x4, 2048, 4MB	[8...8], 2048, 4MB
IPC	1,3063	1,2998
Miss rate	0,0034	0,0032
Accessi L2	24.009.935	24.013.169
Accessi ai banchi	138.610.240	139.095.946
Conflitti	19.864.857	25.213.809

Tabella 30: comparazione di una cache rettangolare da 4MB in modalità interlacciata nelle due configurazioni: D-NUCA 8x4 (test n. 15) e TD-NUCA [8,8,8,8] (test n. 16).

Gli indici sono sostanzialmente invariati ad eccezione del numero di conflitti che è leggermente aumentato per la cache TD-NUCA.

N. test	17	18
Config. L2	D-NUCA	TD-NUCA
Geom. L2	4x4, 2048, 2MB	[4...4], 2048, 2MB
IPC	1,2709	1,2634
Miss rate	0,0155	0,0146
Accessi L2	24.013.849	24.012.488
Accessi ai banchi	154.345.902	156.370.605
Conflitti	48.627.609	50.589.436

Tabella 31: comparazione di una cache rettangolare da 2MB in modalità interlacciata nelle due configurazioni: D-NUCA 4x4 (test n. 17) e TD-NUCA [4,4,4] (test n. 18).

Gli indici sono sostanzialmente invariati ad eccezione del numero di conflitti che è leggermente aumentato per la cache TD-NUCA.

Qui finisce la fase di validazione di 176.gcc. Dai risultati riscontriamo le stesse conclusioni di validazione note dal paragrafo 5.1.

N. test	12	21	22
Config. L2	TD-NUCA	TD-NUCA	TD-NUCA
Geom. L2	[16...16], 1024, 16MB	[2,2,2,2,4,4,4,4,8, 8,8,8,16,16,16,16], 1024, ~8MB	[16,16,16,16,8,8,8, 8,4,4,4,4,2,2,2,2], 1024, ~8MB
IPC	1,1277	1,1699	1,1287
Miss rate	0,0019	0,0019	0,0019
Accessi ai banchi	532.443.533	558.387.712	532.443.499
Via media con hit	1,29	4,54	1,29

Tabella 32: comparazione di una cache rettangolare da 16MB con le equivalenti triangolari crescente e decrescente da circa 8MB.

Per un working set relativamente piccolo (come nel caso di 176.gcc) le caches crescenti danno risultati migliori in quanto presentano latenze di accesso più piccole almeno per le prime vie.

N. test	14	23	24
Config. L2	TD-NUCA	TD-NUCA	TD-NUCA
Geom. L2	[16...16], 1024, 8MB	[2,2,4,4,8,8,16,16], 1024, ~4MB	[16,16,8,8,4,4,2,2], 1024, ~4MB
IPC	1,1297	1,1457	1,1335
Miss rate	0,0019	0,0019	0,0019
Accessi ai banchi	270.013.252	292.268.540	270.043.710
Via media con hit	1,30	4,11	1,25

Tabella 33: comparazione di una cache rettangolare da 8MB con le equivalenti triangolari crescente e decrescente da circa 4MB.

Dato il miss rate molto basso, la cache crescente fornisce ancora risultati migliori rispetto a quella rettangolare.

N. test	16	25	26
Config. L2	TD-NUCA	TD-NUCA	TD-NUCA
Geom. L2	[8...8], 2048, 4MB	[1,2,4,8], 2048, ~2MB	[8,4,2,1], 2048, ~2MB
IPC	1,2998	1,1334	1,3238
Miss rate	0,0032	0,0326	0,0038
Accessi ai banchi	139.095.946	150.204.143	138.925.629
Via media con hit	1,29	3,24	1,17

Tabella 34: comparazione di una cache rettangolare da 4MB con le equivalenti triangolari crescente e decrescente da circa 2MB.

La grandezza del working set è al limite (lo dimostra il miss rate aumentato nella cache crescente). Per questo la cache crescente dà un IPC nettamente più basso. La cache decrescente riesce a fornire prestazioni ancora migliori (2% in più) rispetto alla cache rettangolare.

N. test	18	27	28
Config. L2	TD-NUCA	TD-NUCA	TD-NUCA
Geom. L2	[4...4], 2048, 2MB	[1,1,2,4], 2048, ~1MB	[4,2,1,1], 2048, ~1MB
IPC	1,2634	0,7742	1,1297
Miss rate	0,0146	0,1664	0,0642
Accessi ai banchi	156.370.605	157.700.793	151.637.510
Via media con hit	1,68	3,37	1,33

Tabella 35: comparazione di una cache rettangolare da 2MB con le equivalenti triangolari crescente e decrescente da circa 1MB.

Il miss rate è salito anche per la cache decrescente (6,42%) e questo fa calare le prestazioni (in termini di IPC) del 13% rispetto alla cache rettangolare.

N. test	19	29	30
Config. L2	TD-NUCA	TD-NUCA	TD-NUCA
Geom. L2	[4...4], 1024, 1MB	[1,1,2,4], 1024, ~512KB	[4,2,1,1], 1024, ~512KB
IPC	0,9513	0,4886	0,6418
Miss rate	0,1307	0,4474	0,3916
Accessi ai banchi	162.547.064	163.102.091	153.026.090
Via media con hit	2,24	3,63	1,36

Tabella 36: comparazione di una cache rettangolare da 1MB con le equivalenti triangolari crescente e decrescente da circa 512KB.

Il miss rate è ulteriormente salito per la cache decrescente (39,16%) e questo fa calare le prestazioni (in termini di IPC) del 31% rispetto alla cache rettangolare.

Il comportamento delle prove di base di 176.gcc rispecchia quello della media generale. Questo Spec ha un working set abbastanza ridotto e per questo si ha che la cache triangolare decrescente da 2MB fornisce risultati migliori del 2% rispetto a quella rettangolare da 4MB. Questa tendenza si ha

anche per le cache maggiori, ma non per quelle inferiori con le quali la configurazione triangolare perde il 13% e il 31% sulla configurazione rettangolare da 2MB e da 1MB.

Il grafico 4 riporta gli andamenti dell'IPC al variare delle dimensioni della cache per lo Spec 176.gcc.

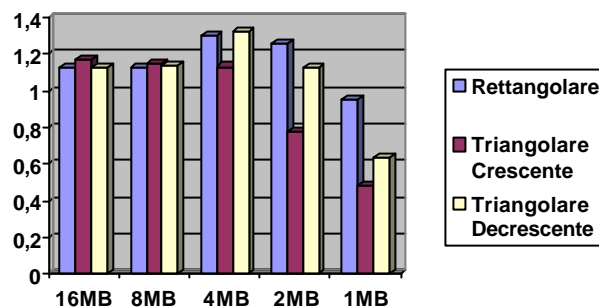


Grafico 4: IPC a confronto per 176.gcc al variare della dimensione delle caches.

Per piccole dimensioni di caches la configurazione triangolare crescente è poco prestante. La configurazione decrescente invece tiene il passo di quella rettangolare almeno fino ai 2MB.

La tendenza ad un IPC migliore per le piccole caches rispetto all'IPC delle caches maggiori si spiega con un ridotto working set di questo Spec che di fatto premia le caches che hanno latenze piccole indipendentemente dalla loro dimensione.

N. test	18	28	31
Config. L2	TD-NUCA	TD-NUCA	TD-NUCA
Geom. L2	[4...4], 2048, 2MB	[4,2,1,1], 2048, ~1MB	[4,2,1,1], 2048, ~1MB
Opzioni attivate	Nessuna	Nessuna	Latenze dei banchi crescenti
IPC	1,2634	1,1297	0,982
Miss rate	0,0146	0,0642	0,0644
Accessi ai banchi	156.370.605	151.637.510	151.529.157

Tabella 37: comparazione della configurazione base di caches triangolari da 2MB con alcune configurazioni avanzate della stessa.

Il ritardo dei banchi di memoria permette di risparmiare il consumo energetico delle caches con un ulteriore peggioramento del 15%.

N. test	32	33	34
Config. L2	TD-NUCA	TD-NUCA	TD-NUCA
Geom. L2	[4,2,1,1], 2048, ~1MB	[4,2,1,1], 2048, ~1MB	[4,2,1,1], 2048, ~1MB
Opzioni attivate	Inserzione casuale	Promozione a 2 vie	Senza promozione
IPC	1,1375	1,069	1,2197
Miss rate	0,0399	0,0865	0,0329
Accessi ai banchi	153.664.891	151.990.280	124.494.755

Tabella 38: comparazione della configurazione base di caches triangolari da 2MB con alcune configurazioni avanzate della stessa (segue).

L'inserzione casuale permette di innalzare l'IPC dell'1% a causa di un minore miss rate (3,99%).

La promozione a 2 vie pessimizza le prestazioni, mentre in assenza di promozione in questo Spec si ha addirittura un miglioramento del 9% con una contemporanea diminuzione nel numero degli accessi ai banchi.

In queste tabelle sono riportati i risultati delle valutazioni delle politiche avanzate per quanto riguarda 176.gcc. Il grafico 5 mostra i loro IPC.

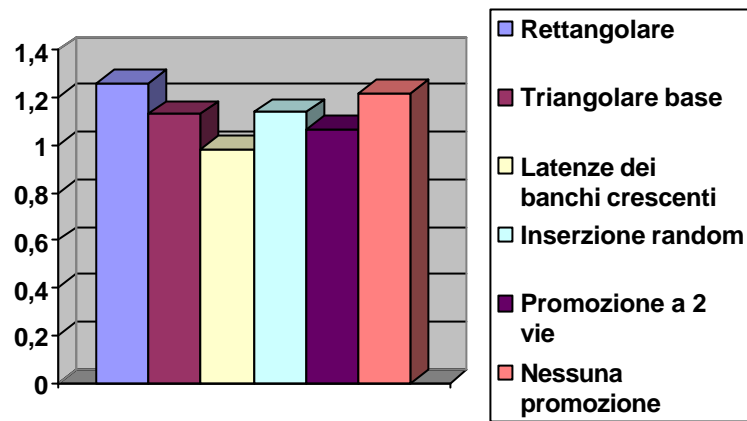


Grafico 5: comparazione degli IPC fra le configurazioni di base ed avanzate per 176.gcc (1MB).

A.2 181.mcf

N. test	1	2
Config. L2	D-NUCA	TD-NUCA
Geom. L2	16x16, 1024, 16MB	[16...16], 1024, 16MB
IPC	0,4741	0,4741
Miss rate	0,0119	0,0119
Accessi L2	40.197.040	40.197.335
Accessi ai banchi	868.869.826	868.874.188
Conflitti	0	0

Tabella 39: comparazione di una cache rettangolare da 16MB in modalità non-interlacciata e con la disabilitazione dei conflitti di canale nelle due configurazioni: D-NUCA 16x16 (test n. 1) e TD-NUCA [16,16,16,16,16,16,16,16,16,16,16,16,16,16,16] (test n. 2).

Tutti gli indici sono perfettamente corrispondenti.

N. test	3	4
Config. L2	D-NUCA	TD-NUCA
Geom. L2	16x8, 1024, 8MB	[16...16], 1024, 8MB
IPC	0,4503	0,4503
Miss rate	0,0292	0,0292
Accessi L2	40.216.110	40.216.259
Accessi ai banchi	458.047.039	458.048.564
Conflitti	0	0

Tabella 40: comparazione di una cache rettangolare da 8MB in modalità non-interlacciata e con la disabilitazione dei conflitti di canale nelle due configurazioni: D-NUCA 16x8 (test n. 3) e TD-NUCA [16,16,16,16,16,16,16,16] (test n. 4).

Tutti gli indici sono perfettamente corrispondenti.

N. test	5	6
Config. L2	D-NUCA	TD-NUCA
Geom. L2	8x4, 2048, 4MB	[8...8], 2048, 4MB
IPC	0,3154	0,3154
Miss rate	0,2105	0,2105
Accessi L2	40.217.351	40.217.245
Accessi ai banchi	241.880.345	241.879.697
Conflitti	0	0

Tabella 41: comparazione di una cache rettangolare da 4MB in modalità non-interlacciata e con la disabilitazione dei conflitti di canale nelle due configurazioni: D-NUCA 8x4 (test n. 5) e TD-NUCA [8,8,8,8] (test n. 6).

Tutti gli indici sono perfettamente corrispondenti.

N. test	7	8
Config. L2	D-NUCA	TD-NUCA
Geom. L2	4x4, 2048, 2MB	[4...4], 2048, 2MB
IPC	0,2505	0,2505
Miss rate	0,4048	0,4048
Accessi L2	40.233.910	40.234.108
Accessi ai banchi	250.307.374	250.305.700
Conflitti	0	0

Tabella 42: comparazione di una cache rettangolare da 2MB in modalità non-interlacciata e con la disabilitazione dei conflitti di canale nelle due configurazioni: D-NUCA 4x4 (test n. 7) e TD-NUCA [4,4,4,4] (test n. 8).

Tutti gli indici sono perfettamente corrispondenti.

Anche per quanto riguarda 181.mcf si può dire che il comportamento delle caches TD-NUCA è in sintonia con quello delle normali caches D-NUCA.

N. test	11	12
Config. L2	D-NUCA	TD-NUCA
Geom. L2	16x16, 1024, 16MB	[16...16], 1024, 16MB
IPC	0,4588	0,4691
Miss rate	0,0119	0,0119
Accessi L2	40.199.183	40.200.140
Accessi ai banchi	868.907.840	869.262.073
Conflitti	61.485.748	191.772.552

Tabella 43: comparazione di una cache rettangolare da 16MB in modalità interlacciata nelle due configurazioni: D-NUCA 16x16 (test n. 11) e TD-NUCA [16,16,16,16,16,16,16,16,16,16,16,16] (test n. 12).

Gli indici sono sostanzialmente invariati ad eccezione del numero di conflitti che è triplicato per la cache TD-NUCA ma che comunque non pregiudica le prestazioni.

N. test	13	14
Config. L2	D-NUCA	TD-NUCA
Geom. L2	16x8, 1024, 8MB	[16...16], 1024, 8MB
IPC	0,4371	0,4463
Miss rate	0,0292	0,0292
Accessi L2	40.217.347	40.219.601
Accessi ai banchi	458.059.687	458.422.238
Conflitti	60.239.805	188.825.461

Tabella 44: comparazione di una cache rettangolare da 8MB in modalità interlacciata nelle due configurazioni: D-NUCA 16x8 (test n. 13) e TD-NUCA [16,16,16,16,16,16,16,16] (test n. 14).

Gli indici sono sostanzialmente invariati ad eccezione del numero di conflitti che è triplicato per la cache TD-NUCA.

N. test	15	16
Config. L2	D-NUCA	TD-NUCA
Geom. L2	8x4, 2048, 4MB	[8...8], 2048, 4MB
IPC	0,312	0,3138
Miss rate	0,2105	0,2105
Accessi L2	40.217.379	40.216.678
Accessi ai banchi	241.882.552	242.740.978
Conflitti	57.845.439	113.128.835

Tabella 45: comparazione di una cache rettangolare da 4MB in modalità interlacciata nelle due configurazioni: D-NUCA 8x4 (test n. 15) e TD-NUCA [8,8,8] (test n. 16).

Gli indici sono sostanzialmente invariati ad eccezione del numero di conflitti che è raddoppiato per la cache TD-NUCA.

N. test	17	18
Config. L2	D-NUCA	TD-NUCA
Geom. L2	4x4, 2048, 2MB	[4...4], 2048, 2MB
IPC	0,2487	0,2496
Miss rate	0,4047	0,4048
Accessi L2	40.227.126	40.234.853
Accessi ai banchi	250.299.295	251.080.335
Conflitti	89.007.365	98.976.057

Tabella 46: comparazione di una cache rettangolare da 2MB in modalità interlacciata nelle due configurazioni: D-NUCA 4x4 (test n. 17) e TD-NUCA [4,4,4] (test n. 18).

Gli indici sono sostanzialmente invariati ad eccezione del numero di conflitti che è leggermente aumentato per la cache TD-NUCA.

Qui finisce la fase di validazione di 181.mcf. Dai risultati riscontriamo le stesse conclusioni di validazione note dal paragrafo 5.1.

N. test	12	21	22
Config. L2	TD-NUCA	TD-NUCA	TD-NUCA
Geom. L2	[16...16], 1024, 16MB	[2,2,2,2,4,4,4,4,8,8,8,8,16,16,16,16], 1024, ~8MB	[16,16,16,16,8,8,8,8,4,4,4,4,2,2,2,2], 1024, ~8MB
IPC	0,4691	0,3744	0,4637
Miss rate	0,0119	0,036	0,0215
Accessi ai banchi	869.262.073	871.180.708	868.951.142
Via media con hit	2,78	10,73	2,41

Tabella 47: comparazione di una cache rettangolare da 16MB con le equivalenti triangolari crescente e decrescente da circa 8MB.

181.mcf ha un working set molto grande (il miss rate per le caches triangolari inizia già ad aumentare) e per questo le prestazioni delle configurazioni triangolari sono peggiori. Il maggior utilizzo della cache da parte di questo Spec si nota anche dalla maggiore media delle vie con hit.

N. test	14	23	24
Config. L2	TD-NUCA	TD-NUCA	TD-NUCA
Geom. L2	[16...16], 1024, 8MB	[2,2,4,4,8,8,16,16], 1024, ~4MB	[16,16,8,8,4,4,2,2], 1024, ~4MB
IPC	0,4463	0,2678	0,3113
Miss rate	0,0292	0,2419	0,1797
Accessi ai banchi	458.422.238	455.737.948	445.724.005
Via media con hit	2,63	6,21	1,72

Tabella 48: comparazione di una cache rettangolare da 8MB con le equivalenti triangolari crescente e decrescente da circa 4MB.

Il miss rate è cresciuto ulteriormente. La cache crescente perde il 18% mentre quella decrescente perde il 14% dell'IPC.

N. test	16	25	26
Config. L2	TD-NUCA	TD-NUCA	TD-NUCA
Geom. L2	[8...8], 2048, 4MB	[1,2,4,8], 2048, ~2MB	[8,4,2,1], 2048, ~2MB
IPC	0,3138	0,2334	0,2512
Miss rate	0,2105	0,4527	0,4246
Accessi ai banchi	242.740.978	251.524.097	238.248.416
Via media con hit	1,91	3,61	1,24

Tabella 49: comparazione di una cache rettangolare da 4MB con le equivalenti triangolari crescente e decrescente da circa 2MB.

La cache crescente perde l'8% mentre quella decrescente perde il 6% dell'IPC.

N. test	18	27	28
Config. L2	TD-NUCA	TD-NUCA	TD-NUCA
Geom. L2	[4...4], 2048, 2MB	[1,1,2,4], 2048, ~1MB	[4,2,1,1], 2048, ~1MB
IPC	0,2496	0,221	0,2282
Miss rate	0,4048	0,536	0,5859
Accessi ai banchi	251.080.335	256.878.010	246.301.295
Via media con hit	2,19	3,54	1,39

Tabella 50: comparazione di una cache rettangolare da 2MB con le equivalenti triangolari crescente e decrescente da circa 1MB.

Il miss rate è estremamente alto e per queste caches di piccole dimensioni causa una perdita di prestazioni del 3% per la configurazione crescente e del 2% per quella decrescente.

N. test	19	29	30
Config. L2	TD-NUCA	TD-NUCA	TD-NUCA
Geom. L2	[4...4], 1024, 1MB	[1,1,2,4], 1024, ~512KB	[4,2,1,1], 1024, ~512KB
IPC	0,2241	0,2107	0,2109
Miss rate	0,5972	0,6778	0,7987
Accessi ai banchi	252.352.685	255.039.978	241.042.875
Via media con hit	2,31	3,58	1,21

Tabella 51: comparazione di una cache rettangolare da 1MB con le equivalenti triangolari crescente e decrescente da circa 512KB.

Il miss rate è ulteriormente salito per la cache decrescente (79,87%) e questo fa calare le prestazioni (in termini di IPC) dell'1% rispetto alla cache rettangolare.

La valutazione della configurazione base per 181.mcf ci conferma la tendenza generale riscontrata nel capitolo 5. Essendo però il working set di questo Spec molto grande il calo di prestazioni delle caches avviene già per le caches da 8MB anticipando la tendenza generale.

Il grafico 6 mostra l'andamento degli IPC per le diverse caches. A causa di un working set corposo le prestazioni delle caches di piccole dimensioni sono fortemente ridotte rispetto a quelle delle caches più grandi.

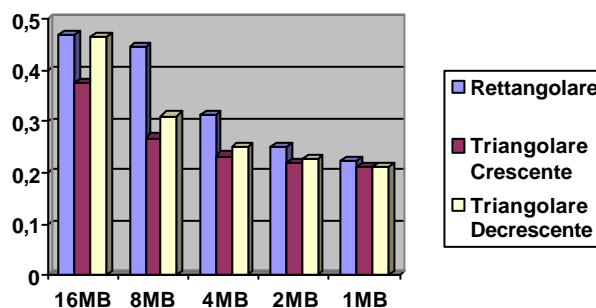


Grafico 6: IPC a confronto per 181.mcf al variare della dimensione delle caches.

N. test	18	28	31
Config. L2	TD-NUCA	TD-NUCA	TD-NUCA
Geom. L2	[4...4], 2048, 2MB	[4,2,1,1], 2048, ~1MB	[4,2,1,1], 2048, ~1MB
Opzioni attivate	Nessuna	Nessuna	Latenze dei banchi crescenti
IPC	0,2496	0,2282	0,2032
Miss rate	0,4048	0,5859	0,5896
Accessi ai banchi	251.080.335	246.301.295	246.167.171

Tabella 52: comparazione della configurazione base di caches triangolari da 2MB con alcune configurazioni avanzate della stessa.

Il ritardo dei banchi di memoria permette di risparmiare il consumo energetico delle caches con un ulteriore peggioramento del 3%.

N. test	32	33	34
Config. L2	TD-NUCA	TD-NUCA	TD-NUCA
Geom. L2	[4,2,1,1], 2048, ~1MB	[4,2,1,1], 2048, ~1MB	[4,2,1,1], 2048, ~1MB
Opzioni attivate	Inserzione casuale	Promozione a 2 vie	Senza promozione
IPC	0,2153	0,2248	0,2228
Miss rate	0,5683	0,6113	0,637
Accessi ai banchi	281.810.002	246.460.912	244.532.140

Tabella 53: comparazione della configurazione base di caches triangolari da 2MB con alcune configurazioni avanzate della stessa (segue).

L'inserzione casuale permette di innalzare l'IPC dell'1% a causa di un minore miss rate.

Per questo Spec sia la soluzione a doppia promozione che quella senza promozione non danno risultati apprezzabili, infatti sia i loro IPC che il loro numero di accessi ai banchi è rimasto pressoché costante.

Nelle tabelle 52 e 53 sono riportati i risultati delle simulazioni per le configurazioni avanzate. Il grafico 7 riassume le loro prestazioni.

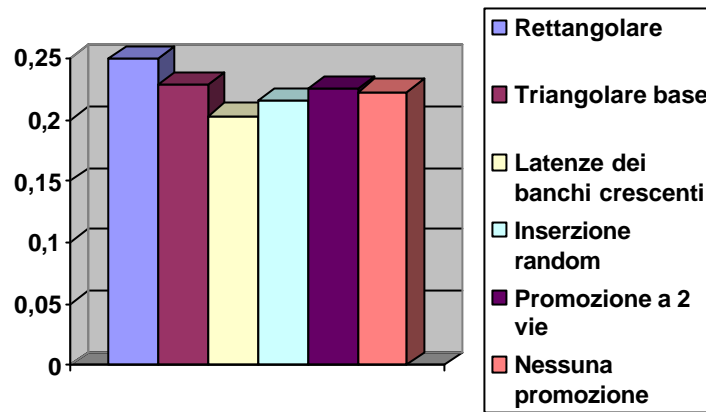


Grafico 7: comparazione degli IPC fra le configurazioni di base ed avanzate per 181.mcf (1MB).

A.3 256.bzip2

N. test	1	2
Config. L2	D-NUCA	TD-NUCA
Geom. L2	16x16, 1024, 16MB	[16...16], 1024, 16MB
IPC	1,3836	1,3836
Miss rate	0,0119	0,0119
Accessi L2	9.245.837	9.245.838
Accessi ai banchi	186.677.754	186.677.754
Conflitti	0	0

Tabella 54: comparazione di una cache rettangolare da 16MB in modalità non-interlacciata e con la disabilitazione dei conflitti di canale nelle due configurazioni: D-NUCA 16x16 (test n. 1) e TD-NUCA [16,16,16,16,16,16,16,16,16,16,16,16,16,16,16] (test n. 2).

Tutti gli indici sono perfettamente corrispondenti.

N. test	3	4
Config. L2	D-NUCA	TD-NUCA
Geom. L2	16x8, 1024, 8MB	[16...16], 1024, 8MB
IPC	1,3851	1,3851
Miss rate	0,0119	0,0119
Accessi L2	9.246.086	9.246.087
Accessi ai banchi	95.383.762	95.383.762
Conflitti	0	0

Tabella 55: comparazione di una cache rettangolare da 8MB in modalità non-interlacciata e con la disabilitazione dei conflitti di canale nelle due configurazioni: D-NUCA 16x8 (test n. 3) e TD-NUCA [16,16,16,16,16,16,16,16] (test n. 4).

Tutti gli indici sono perfettamente corrispondenti.

N. test	5	6
Config. L2	D-NUCA	TD-NUCA
Geom. L2	8x4, 2048, 4MB	[8...8], 2048, 4MB
IPC	1,4029	1,403
Miss rate	0,03	0,03
Accessi L2	9.245.167	9.245.259
Accessi ai banchi	50.028.336	50.028.764
Conflitti	0	0

Tabella 56: comparazione di una cache rettangolare da 4MB in modalità non-interlacciata e con la disabilitazione dei conflitti di canale nelle due configurazioni: D-NUCA 8x4 (test n. 5) e TD-NUCA [8,8,8,8] (test n. 6).

Tutti gli indici sono perfettamente corrispondenti.

N. test	7	8
Config. L2	D-NUCA	TD-NUCA
Geom. L2	4x4, 2048, 2MB	[4...4], 2048, 2MB
IPC	1,3791	1,3793
Miss rate	0,0834	0,0834
Accessi L2	9.244.954	9.245.499
Accessi ai banchi	54.097.893	54.098.597
Conflitti	0	0

Tabella 57: comparazione di una cache rettangolare da 2MB in modalità non-interlacciata e con la disabilitazione dei conflitti di canale nelle due configurazioni: D-NUCA 4x4 (test n. 7) e TD-NUCA [4,4,4,4] (test n. 8).

Tutti gli indici sono perfettamente corrispondenti.

256.bzip2 mostra un analogo comportamento fra le due configurazioni di caches (D-NUCA e TD-NUCA).

N. test	11	12
Config. L2	D-NUCA	TD-NUCA
Geom. L2	16x16, 1024, 16MB	[16...16], 1024, 16MB
IPC	1,3884	1,3895
Miss rate	0,0119	0,0119
Accessi L2	9.246.064	9.246.099
Accessi ai banchi	186.680.489	186.714.211
Conflitti	8.388.176	15.996.565

Tabella 58: comparazione di una cache rettangolare da 16MB in modalità interlacciata nelle due configurazioni: D-NUCA 16x16 (test n. 11) e TD-NUCA [16,16,16,16,16,16,16,16,16,16,16,16,16,16] (test n. 12).

Gli indici sono sostanzialmente invariati ad eccezione del numero di conflitti che è raddoppiato per la cache TD-NUCA ma che comunque non pregiudica le prestazioni.

N. test	13	14
Config. L2	D-NUCA	TD-NUCA
Geom. L2	16x8, 1024, 8MB	[16...16], 1024, 8MB
IPC	1,3897	1,3911
Miss rate	0,0119	0,0119
Accessi L2	9.246.091	9.246.167
Accessi ai banchi	95.383.182	95.416.998
Conflitti	8.392.319	16.026.317

Tabella 59: comparazione di una cache rettangolare da 8MB in modalità interlacciata nelle due configurazioni: D-NUCA 16x8 (test n. 13) e TD-NUCA [16,16,16,16,16,16,16,16,16,16,16,16,16,16] (test n. 14).

Gli indici sono sostanzialmente invariati ad eccezione del numero di conflitti che è raddoppiato per la cache TD-NUCA.

N. test	15	16
Config. L2	D-NUCA	TD-NUCA
Geom. L2	8x4, 2048, 4MB	[8...8], 2048, 4MB
IPC	1,4041	1,4047
Miss rate	0,0301	0,03
Accessi L2	9.245.347	9.245.059
Accessi ai banchi	50.029.701	50.112.018
Conflitti	9.301.990	12.094.767

Tabella 60: comparazione di una cache rettangolare da 4MB in modalità interlacciata nelle due configurazioni: D-NUCA 8x4 (test n. 15) e TD-NUCA [8,8,8] (test n. 16).

Gli indici sono sostanzialmente invariati ad eccezione del numero di conflitti che è leggermente aumentato per la cache TD-NUCA.

N. test	17	18
Config. L2	D-NUCA	TD-NUCA
Geom. L2	4x4, 2048, 2MB	[4...4], 2048, 2MB
IPC	1,3759	1,3758
Miss rate	0,0834	0,0834
Accessi L2	9.245.237	9.245.520
Accessi ai banchi	54.099.585	54.301.869
Conflitti	13.401.271	14.280.706

Tabella 61: comparazione di una cache rettangolare da 2MB in modalità interlacciata nelle due configurazioni: D-NUCA 4x4 (test n. 17) e TD-NUCA [4,4,4] (test n. 18).

Gli indici sono sostanzialmente invariati ad eccezione del numero di conflitti che è leggermente aumentato per la cache TD-NUCA.

Da questo momento anche 256.bzip2 ha confermato la validazione delle due caches data la corrispondenza delle loro prestazioni.

N. test	12	21	22
Config. L2	TD-NUCA	TD-NUCA	TD-NUCA
Geom. L2	[16...16], 1024, 16MB	[2,2,2,2,4,4,4,4,8,8,8,8,16,16,16,16], 1024, ~8MB	[16,16,16,16,8,8,8,8,4,4,4,4,2,2,2,2], 1024, ~8MB
IPC	1,3895	1,3739	1,3906
Miss rate	0,0119	0,012	0,0119
Accessi ai banchi	186.714.211	193.345.311	186.729.780
Via media con hit	1,45	8,70	1,43

Tabella 62: comparazione di una cache rettangolare da 16MB con le equivalenti triangolari crescente e decrescente da circa 8MB.

256.bzip2 ha un working set abbastanza piccolo e per questo le tre configurazioni di caches hanno prestazioni abbastanza allineate.

N. test	14	23	24
Config. L2	TD-NUCA	TD-NUCA	TD-NUCA
Geom. L2	[16...16], 1024, 8MB	[2,2,4,4,8,8,16,16], 1024, ~4MB	[16,16,8,8,4,4,2,2], 1024, ~4MB
IPC	1,3911	1,3673	1,3728
Miss rate	0,0119	0,0359	0,043
Accessi ai banchi	95.416.998	100.377.195	96.356.169
Via media con hit	1,45	5,75	1,31

Tabella 63: comparazione di una cache rettangolare da 8MB con le equivalenti triangolari crescente e decrescente da circa 4MB.

Il miss rate inizia ad aumentare causando una perdita del 2% sull'IPC sia della cache crescente sia di quella decrescente.

N. test	16	25	26
Config. L2	TD-NUCA	TD-NUCA	TD-NUCA
Geom. L2	[8...8], 2048, 4MB	[1,2,4,8], 2048, ~2MB	[8,4,2,1], 2048, ~2MB
IPC	1,4047	1,3439	1,3573
Miss rate	0,03	0,1174	0,118
Accessi ai banchi	50.112.018	51.704.813	50.388.459
Via media con hit	1,38	3,49	1,19

Tabella 64: comparazione di una cache rettangolare da 4MB con le equivalenti triangolari crescente e decrescente da circa 2MB.

Il miss rate delle caches triangolari è al 12%. La cache crescente dà un IPC del 6% più basso. La cache decrescente perde il 5%.

N. test	18	27	28
Config. L2	TD-NUCA	TD-NUCA	TD-NUCA
Geom. L2	[4...4], 2048, 2MB	[1,1,2,4], 2048, ~1MB	[4,2,1,1], 2048, ~1MB
IPC	1,3758	1,2942	1,3015
Miss rate	0,0834	0,239	0,2987
Accessi ai banchi	54.301.869	54.971.215	53.730.170
Via media con hit	1,77	3,46	1,28

Tabella 65: comparazione di una cache rettangolare da 2MB con le equivalenti triangolari crescente e decrescente da circa 1MB.

Il miss rate della cache decrescente è salito al 30% e questo fa calare le prestazioni (in termini di IPC) del 7% rispetto alla cache rettangolare.

N. test	19	29	30
Config. L2	TD-NUCA	TD-NUCA	TD-NUCA
Geom. L2	[4...4], 1024, 1MB	[1,1,2,4], 1024, ~512KB	[4,2,1,1], 1024, ~512KB
IPC	1,3022	1,2192	1,2058
Miss rate	0,2393	0,4239	0,602
Accessi ai banchi	57.429.394	56.069.883	54.776.573
Via media con hit	2,09	3,57	1,34

Tabella 66: comparazione di una cache rettangolare da 1MB con le equivalenti triangolari crescente e decrescente da circa 512KB.

Il miss rate è ulteriormente salito per la cache decrescente (60,20%) e questo fa calare le prestazioni (in termini di IPC) del 10% rispetto alla cache rettangolare.

Anche per 256.bzip2 la caches crescenti si comportano sempre in modo peggiore di quelle decrescenti. Inoltre queste ultime riescono a dare prestazioni equivalenti alle rettangolari almeno fino alla dimensione di 4MB per la quale il working set risulta maggiore della dimensione delle caches triangolari. Il grafico 8 riassume queste considerazioni.

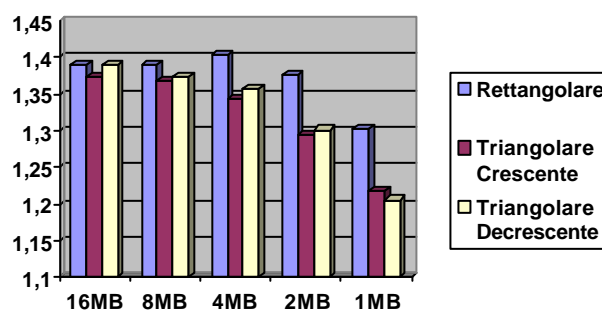


Grafico 8: IPC a confronto per 256.bzip2 al variare della dimensione delle caches.

N. test	18	28	31
Config. L2	TD-NUCA	TD-NUCA	TD-NUCA
Geom. L2	[4...4], 2048, 2MB	[4,2,1,1], 2048, ~1MB	[4,2,1,1], 2048, ~1MB
Opzioni attivate	Nessuna	Nessuna	Latenze dei banchi crescenti
IPC	1,3758	1,3015	1,2788
Miss rate	0,0834	0,2987	0,3037
Accessi ai banchi	54.301.869	53.730.170	53.660.614

Tabella 67: comparazione della configurazione base di caches triangolari da 2MB con alcune configurazioni avanzate della stessa.

Il ritardo dei banchi di memoria permette di risparmiare il consumo energetico delle caches con un ulteriore peggioramento del 2%.

N. test	32	33	34
Config. L2	TD-NUCA	TD-NUCA	TD-NUCA
Geom. L2	[4,2,1,1], 2048, ~1MB	[4,2,1,1], 2048, ~1MB	[4,2,1,1], 2048, ~1MB
Opzioni attivate	Inserzione casuale	Promozione a 2 vie	Senza promozione
IPC	1,3077	1,2878	1,2102
Miss rate	0,2198	0,3197	0,5573
Accessi ai banchi	56.261.432	53.281.498	51.274.780

Tabella 68: comparazione della configurazione base di caches triangolari da 2MB con alcune configurazioni avanzate della stessa (segue).

L'inserzione casuale permette di innalzare l'IPC dell'1% a causa di un minore miss rate.

La promozione a 2 vie peggiora le prestazioni dell'1%, mentre in assenza di promozione le prestazioni vengono notevolmente ridotte.

Il grafico 9 mostra i risultati delle simulazioni avanzate per 256.bzip2. Un comportamento anomalo è dato dalla riduzione delle prestazioni della configurazione senza promozione: probabilmente questo benchmark ha una notevole località e l'assenza di una politica LRU lo penalizza molto.

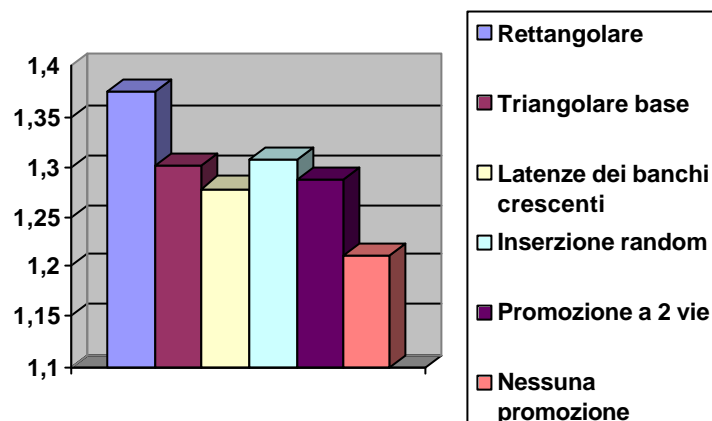


Grafico 9: comparazione degli IPC fra le configurazioni di base ed avanzate per 256.bzip2 (1 MB).

A.4 300.twolf

N. test	1	2
Config. L2	D-NUCA	TD-NUCA
Geom. L2	16x16, 1024, 16MB	[16...16], 1024, 16MB
IPC	0,9185	0,9185
Miss rate	0,0046	0,0046
Accessi L2	5.846.090	5.846.092
Accessi ai banchi	120.562.154	120.562.154
Conflitti	0	0

Tabella 69: comparazione di una cache rettangolare da 16MB in modalità non-interlacciata e con la disabilitazione dei conflitti di canale nelle due configurazioni: D-NUCA 16x16 (test n. 1) e TD-NUCA [16,16,16,16,16,16,16,16,16,16,16,16,16,16,16] (test n. 2).

Tutti gli indici sono perfettamente corrispondenti.

N. test	3	4
Config. L2	D-NUCA	TD-NUCA
Geom. L2	16x8, 1024, 8MB	[16...16], 1024, 8MB
IPC	0,9199	0,9199
Miss rate	0,0046	0,0046
Accessi L2	5.846.123	5.846.125
Accessi ai banchi	61.148.323	61.148.323
Conflitti	0	0

Tabella 70: comparazione di una cache rettangolare da 8MB in modalità non-interlacciata e con la disabilitazione dei conflitti di canale nelle due configurazioni: D-NUCA 16x8 (test n. 3) e TD-NUCA [16,16,16,16,16,16,16,16] (test n. 4).

Tutti gli indici sono perfettamente corrispondenti.

N. test	5	6
Config. L2	D-NUCA	TD-NUCA
Geom. L2	8x4, 2048, 4MB	[8...8], 2048, 4MB
IPC	0,9702	0,9702
Miss rate	0,0046	0,0046
Accessi L2	5.842.004	5.842.006
Accessi ai banchi	31.423.548	31.423.548
Conflitti	0	0

Tabella 71: comparazione di una cache rettangolare da 4MB in modalità non-interlacciata e con la disabilitazione dei conflitti di canale nelle due configurazioni: D-NUCA 8x4 (test n. 5) e TD-NUCA [8,8,8,8] (test n. 6).

Tutti gli indici sono perfettamente corrispondenti.

N. test	7	8
Config. L2	D-NUCA	TD-NUCA
Geom. L2	4x4, 2048, 2MB	[4...4], 2048, 2MB
IPC	0,9868	0,9868
Miss rate	0,0046	0,0046
Accessi L2	5.866.624	5.866.627
Accessi ai banchi	34.549.637	34.549.645
Conflitti	0	0

Tabella 72: comparazione di una cache rettangolare da 2MB in modalità non-interlacciata e con la disabilitazione dei conflitti di canale nelle due configurazioni: D-NUCA 4x4 (test n. 7) e TD-NUCA [4,4,4,4] (test n. 8).

Tutti gli indici sono perfettamente corrispondenti.

Tutti gli indici sono esattamente corrispondenti inducendoci a concludere lo stesso comportamento delle due configurazioni di caches.

N. test	11	12
Config. L2	D-NUCA	TD-NUCA
Geom. L2	16x16, 1024, 16MB	[16...16], 1024, 16MB
IPC	0,927	0,9286
Miss rate	0,0046	0,0046
Accessi L2	5.881.071	5.880.561
Accessi ai banchi	121.157.486	121.156.580
Conflitti	7.578.869	8.848.425

Tabella 73: comparazione di una cache rettangolare da 16MB in modalità interlacciata nelle due configurazioni: D-NUCA 16x16 (test n. 11) e TD-NUCA [16,16,16,16,16,16,16,16,16,16,16,16] (test n. 12).

Gli indici sono sostanzialmente invariati. Anche il numero di conflitti è rimasto pressoché invariato.

N. test	13	14
Config. L2	D-NUCA	TD-NUCA
Geom. L2	16x8, 1024, 8MB	[16...16], 1024, 8MB
IPC	0,9283	0,9299
Miss rate	0,0046	0,0046
Accessi L2	5.881.140	5.880.566
Accessi ai banchi	61.453.687	61.456.611
Conflitti	7.582.456	8.852.723

Tabella 74: comparazione di una cache rettangolare da 8MB in modalità interlacciata nelle due configurazioni: D-NUCA 16x8 (test n. 13) e TD-NUCA [16,16,16,16,16,16,16,16] (test n. 14).

Gli indici sono sostanzialmente invariati.

N. test	15	16
Config. L2	D-NUCA	TD-NUCA
Geom. L2	8x4, 2048, 4MB	[8...8], 2048, 4MB
IPC	0,9695	0,9705
Miss rate	0,0046	0,0046
Accessi L2	5.874.187	5.873.924
Accessi ai banchi	31.571.133	31.597.835
Conflitti	7.728.045	6.841.430

Tabella 75: comparazione di una cache rettangolare da 4MB in modalità interlacciata nelle due configurazioni: D-NUCA 8x4 (test n. 15) e TD-NUCA [8,8,8] (test n. 16).

Gli indici sono sostanzialmente invariati ad eccezione del numero di conflitti che è leggermente aumentato per la cache TD-NUCA.

N. test	17	18
Config. L2	D-NUCA	TD-NUCA
Geom. L2	4x4, 2048, 2MB	[4...4], 2048, 2MB
IPC	0,9776	0,9779
Miss rate	0,0046	0,0046
Accessi L2	5.877.716	5.878.411
Accessi ai banchi	34.612.859	34.742.992
Conflitti	10.048.252	8.578.638

Tabella 76: comparazione di una cache rettangolare da 2MB in modalità interlacciata nelle due configurazioni: D-NUCA 4x4 (test n. 17) e TD-NUCA [4,4,4] (test n. 18).

Gli indici sono sostanzialmente invariati ad eccezione del numero di conflitti che è addirittura diminuito per la cache TD-NUCA.

Qui si conclude positivamente la fase di validazione di 300.twolf.

N. test	12	21	22
Config. L2	TD-NUCA	TD-NUCA	TD-NUCA
Geom. L2	[16...16], 1024, 16MB	[2,2,2,2,4,4,4,4,8,8,8,8,16,16,16,16], 1024, ~8MB	[16,16,16,16,8,8,8,8,4,4,4,4,2,2,2,2], 1024, ~8MB
IPC	0,9286	0,9288	0,9291
Miss rate	0,0046	0,0046	0,0046
Accessi ai banchi	121.156.580	126.050.659	121.156.926
Via media con hit	1,20	4,95	1,20

Tabella 77: comparazione di una cache rettangolare da 16MB con le equivalenti triangolari crescente e decrescente da circa 8MB.

Per un working set relativamente piccolo (come nel caso di questo benchmark) le caches triangolari danno risultati analoghi a quelli della cache rettangolare.

N. test	14	23	24
Config. L2	TD-NUCA	TD-NUCA	TD-NUCA
Geom. L2	[16...16], 1024, 8MB	[2,2,4,4,8,8,16,16], 1024, ~4MB	[16,16,8,8,4,4,2,2], 1024, ~4MB
IPC	0,9299	0,934	0,9304
Miss rate	0,0046	0,0046	0,0046
Accessi ai banchi	61.456.611	64.874.223	61.456.345
Via media con hit	1,20	4,31	1,20

Tabella 78: comparazione di una cache rettangolare da 8MB con le equivalenti triangolari crescente e decrescente da circa 4MB.

Dato il miss rate ancora molto basso (0,46%) le prestazioni dei tre tests sono ancora molto vicine.

N. test	16	25	26
Config. L2	TD-NUCA	TD-NUCA	TD-NUCA
Geom. L2	[8...8], 2048, 4MB	[1,2,4,8], 2048, ~2MB	[8,4,2,1], 2048, ~2MB
IPC	0,9705	0,9026	0,9782
Miss rate	0,0046	0,0415	0,0046
Accessi ai banchi	31.597.835	32.712.832	31.476.521
Via media con hit	1,20	3,38	1,14

Tabella 79: comparazione di una cache rettangolare da 4MB con le equivalenti triangolari crescente e decrescente da circa 2MB.

La grandezza del working set è al limite (lo dimostra il miss rate aumentato nella cache crescente). Per questo la cache crescente dà un IPC nettamente più basso. La cache decrescente riesce a fornire prestazioni ancora simili rispetto alla cache rettangolare.

N. test	18	27	28
Config. L2	TD-NUCA	TD-NUCA	TD-NUCA
Geom. L2	[4...4], 2048, 2MB	[1,1,2,4], 2048, ~1MB	[4,2,1,1], 2048, ~1MB
IPC	0,9779	0,7755	0,8368
Miss rate	0,0046	0,1795	0,1214
Accessi ai banchi	34.742.992	34.895.121	33.842.094
Via media con hit	1,62	3,39	1,24

Tabella 80: comparazione di una cache rettangolare da 2MB con le equivalenti triangolari crescente e decrescente da circa 1MB.

Il miss rate è salito anche per la cache decrescente (12,14%) e questo fa calare le prestazioni (in termini di IPC) del 14% rispetto alla cache rettangolare.

N. test	19	29	30
Config. L2	TD-NUCA	TD-NUCA	TD-NUCA
Geom. L2	[4...4], 1024, 1MB	[1,1,2,4], 1024, ~512KB	[4,2,1,1], 1024, ~512KB
IPC	0,8208	0,6478	0,6841
Miss rate	0,1285	0,3907	0,3599
Accessi ai banchi	36.244.521	35.966.922	34.546.507
Via media con hit	1,96	3,45	1,26

Tabella 81: comparazione di una cache rettangolare da 1MB con le equivalenti triangolari crescente e decrescente da circa 512KB.

Il miss rate è ulteriormente salito per la cache decrescente (35,99%) e questo fa calare le prestazioni (in termini di IPC) del 14% rispetto alla cache rettangolare.

Il grafico 10 mostra l'andamento dell'IPC dei tre tipi di caches al variare della loro dimensione per 300.twolf. Le caches di piccole dimensioni riescono a dare un IPC migliore di quello relativo alle caches di grandi dimensioni, questo a causa di un working set abbastanza piccolo di 300.twolf. Solo per l'ultima serie di tests le prestazioni della cache decrescente risultano peggiori (del 14%) rispetto alla relativa rettangolare.

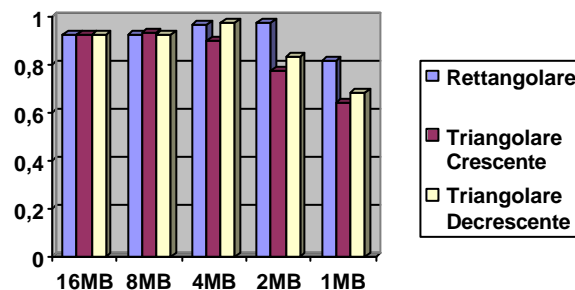


Grafico 10: IPC a confronto per 300.twolf al variare della dimensione delle caches.

N. test	18	28	31
Config. L2	TD-NUCA	TD-NUCA	TD-NUCA
Geom. L2	[4...4], 2048, 2MB	[4,2,1,1], 2048, ~1MB	[4,2,1,1], 2048, ~1MB
Opzioni attivate	Nessuna	Nessuna	Latenze dei banchi crescenti
IPC	0,9779	0,8368	0,8199
Miss rate	0,0046	0,1214	0,1213
Accessi ai banchi	34.742.992	33.842.094	33.881.920

Tabella 82: comparazione della configurazione base di caches triangolari da 2MB con alcune configurazioni avanzate della stessa.

Il ritardo dei banchi di memoria permette di risparmiare il consumo energetico delle caches con un ulteriore peggioramento del 2%.

N. test	32	33	34
Config. L2	TD-NUCA	TD-NUCA	TD-NUCA
Geom. L2	[4,2,1,1], 2048, ~1MB	[4,2,1,1], 2048, ~1MB	[4,2,1,1], 2048, ~1MB
Opzioni attivate	Inserzione casuale	Promozione a 2 vie	Senza promozione
IPC	0,8190	0,8250	0,7464
Miss rate	0,1379	0,1337	0,2568
Accessi ai banchi	35.301.517	33.996.965	31.351.350

Tabella 83: comparazione della configurazione base di caches triangolari da 2MB con alcune configurazioni avanzate della stessa (segue).

L'inserzione casuale riduce l'IPC del 2% a causa di un maggiore miss rate.

La promozione a 2 vie sostanzialmente non varia le prestazioni, mentre in assenza di promozione in questo Spec si ha un peggioramento del 9% delle stesse.

Il grafico 11 mostra i risultati delle valutazioni delle configurazioni avanzate. Si riscontra anche in questo benchmark un calo delle prestazioni per la configurazione con assenza di promozione. La configurazione del test

31 pur permettendo un forte calo del consumo pregiudica le prestazioni solo del 2%.

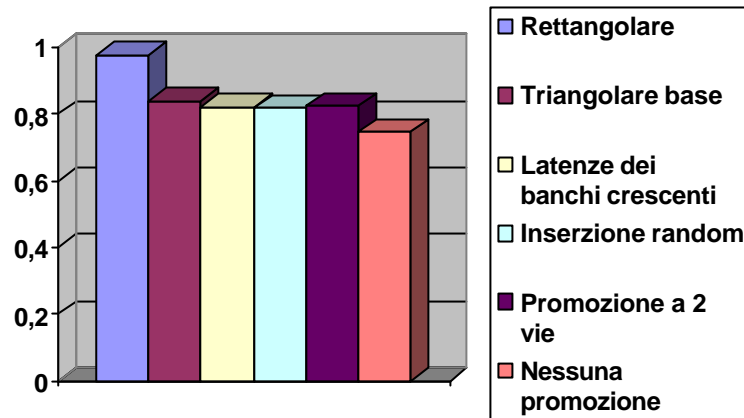


Grafico 11: comparazione degli IPC fra le configurazioni di base ed avanzate per 300.twolf (1MB).

RIFERIMENTI

- [1] R. Desikan, D. Burger, S. Keckler, T. Austin, "Sim-Alpha: a Validated, Execution-Driven Alpha 21264 Simulator".
- [2] C. Kim, D. Burger, S. Keckler, "An Adaptive, Non-Uniform Cache Structure for Wire-Delay Dominated On-Chip Caches", ASPLOS.
- [3] International Technology Roadmap for Semiconductor, 2001.
- [4] J. L. Cruz, A. González, M. Valero, N. Topham, "Multiple-Banked Register File Architectures", ISCA 2000.
- [5] A. Sakamata, T. Sato, "A Leakage-Energy-Reduction Technique for High-Associativity Caches in Embedded Systems".
- [6] C. Kim, D. Burger, S. Keckler, "An Adaptive Cache Structure for Future High-Performance Systems".
- [7] D. Burger, T. Austin, "The Simple Scalar Tool Set, Version 2.0".
- [8] "Intel Pentium 4 Processor in the 478-Pin Package at 1.40 GHz, 1.50 GHz, 1.60GHz, 1.70 GHz, 1.80 GHz, 1.90GHz, and 2.00 GHz".
- [9] "AMD Athlon XP Processor Model 10 Data Sheet", may 2003.
- [10] Compaq Computer Corporation, "21264/EV68CB and 21264/EV68DC Hardware Reference Manual", pp 39-40 and 87-94, june 2001.
- [11] Compaq Computer Corporation, "Alpha 21264 Microprocessor Hardware Reference Manual".
- [12] "SIA Annual Report 2003".
- [13] A. M. del Corral, J. M. Llaberia, "Eliminating Conflicts between Vector Streams in Interleaved Memory Systems".
- [14] A. M. del Corral, J. M. Llaberia, "Who to Increase the Effective Memory Bandwidth in Multivector Processors".

- [15] W. Oed, O. Lange, “On the Effective Bandwidth of Interleaved Memories in Vector Processor System”.
- [16] Semiconductor Industry Association, <http://www.semichips.org>.
- [17] Standard Performance Evaluation Corporation, <http://www.specbench.org>.

BIBLIOGRAFIA

- [18] C. Cowell, C. A. Moritz, W. Burleson, “Improved Modelling and Data Migration for Dynamic Non-Uniform Cache Access”.
- [19] R. Desikan, D. Burger, S. Keckler, “Measuring Experimental Error in Microprocessor Simulation”, ISCA 2001.
- [20] J. M. Parcerisa, A. González, “Reducing Wire Delay Penalty through Value Prediction”, ISM 2000.
- [21] H. Hanson, S. W. Keckler, D. Burger, “Static Energy Reduction Techniques in Microprocessor Caches”.
- [22] T. Karkhanis, J. E. Smith, “A Day in the Life of a Data Cache Miss”.
- [23] A. J. Smith, “Cache Memories”.
- [24] H. Hanson, M.S. Hrishikesh, V. Agarwal, S. W. Keckler, D. Burger, “Static Energy Reduction Techniques for Microprocessor Caches”.
- [25] D. Burger, A. Kagi, M.S. Hrishikesh, “Memory Hierarchy Extension to the SimpleScalar Tool Set”.
- [26] Lizy Kurian John, “Data Placement Schemes to Reduce Conflicts in Interleaved Memories”.
- [27] M. Peiron, M. Valero, E. Ayguadé, T. Lang, “HwA2: Special Memory Interleaving Schemes”.