UNIVERSITAT POLITÈCNICA DE CATALUNYA

Departament d'Arquitectura de Computadors

RECURSOS ANCHOS: UNA TÉCNICA DE BAJO COSTE PARA EXPLOTAR PARALELISMO AGRESIVO EN CÓDIGOS NUMÉRICOS

Autor: David López Alvarez
Directores: Mateo Valero Cortés
Josep Llosa i Espuny

Capítulo 7.

Escogiendo un modelo: relación rendimiento/coste

7.1 Introducción

En el capítulo 5 estudiamos el rendimiento teórico de las técnicas de *widening*, replicación y fusión de operaciones multiplicación y suma (uso de unidades funcionales capaces de implementar la operación FMA). También estudiamos el rendimiento una vez efectuada una planificación con un banco de registros finito, y vimos cómo las configuraciones con mejor rendimiento teórico no eran necesariamente las que ofrecían mejor rendimiento una vez realizada dicha planificación. Se planteó la necesidad de calcular el coste de las configuraciones para averiguar cuál era la mejor, y en el capítulo 6 vimos cómo calcular dicho coste.

En este capítulo vamos a estudiar qué configuraciones ofrecen un mejor rendimiento cuando trabajamos bajo un límite tecnológico. La metodología seguida es la siguiente: primero seleccionamos qué configuraciones pueden ser implementables de acuerdo con las previsiones de la SIA y siguiendo el modelo de área expuesto en la sección 6.2. Las configuraciones siguen

la notación **XwY(Z:n)**, donde se tiene **X** buses y 2*X FPUs, todos de ancho **Y**; asimismo se dispone de un banco de **Z** registros de ancho **Y**, particionado en **n** bloques. Para cada configuración, se calcula su tiempo de ciclo, asumiendo que está limitado por el tiempo de ciclo del banco de registros. La latencia de las unidades funcionales se adapta a su vez al tiempo de ciclo. Finalmente se calcula el número de ciclos necesario para ejecutar cada bucle, lo que, junto con el tiempo de ciclo, nos da el rendimiento final.

7.2 Configuraciones implementables.

El precio es lo que pagas,
El valor, lo que obtienes.

Anónimo

Dada una tecnología, el área que podemos dedicar al banco de registros y las FPUs está limitada, por lo que el número de configuraciones implementables para cada generación de las propuestas por la SIA viene dado por el coste en área de dichas configuraciones.

En la figura 7.1 se muestra el coste en área de diversas configuraciones con bancos de 32, 64, 128 y 256 registros, para configuraciones que no implementan FMA (las barras de color) y configuraciones que implementan FMA (la extensión blanca).

Como ya hemos dicho, consideramos que un procesador suele dedicar entre el 10% y el 20% de su área a las FPUs y al banco de registros asociado (los cuadros en línea discontinua). Por tanto, consideramos que una configuración es implementable bajo una tecnología si la suma de las áreas de las FPUs y el banco de registros ocupa menos del 20% del área total del *chip*, área calculada según las previsiones de la SIA.

La figura 7.1 muestra el coste en área para un banco de registros no particionado. Como ya vimos en la sección 6.3, el particionado decrementa el tiempo de ciclo del banco de registros, pero incrementando su coste en área.

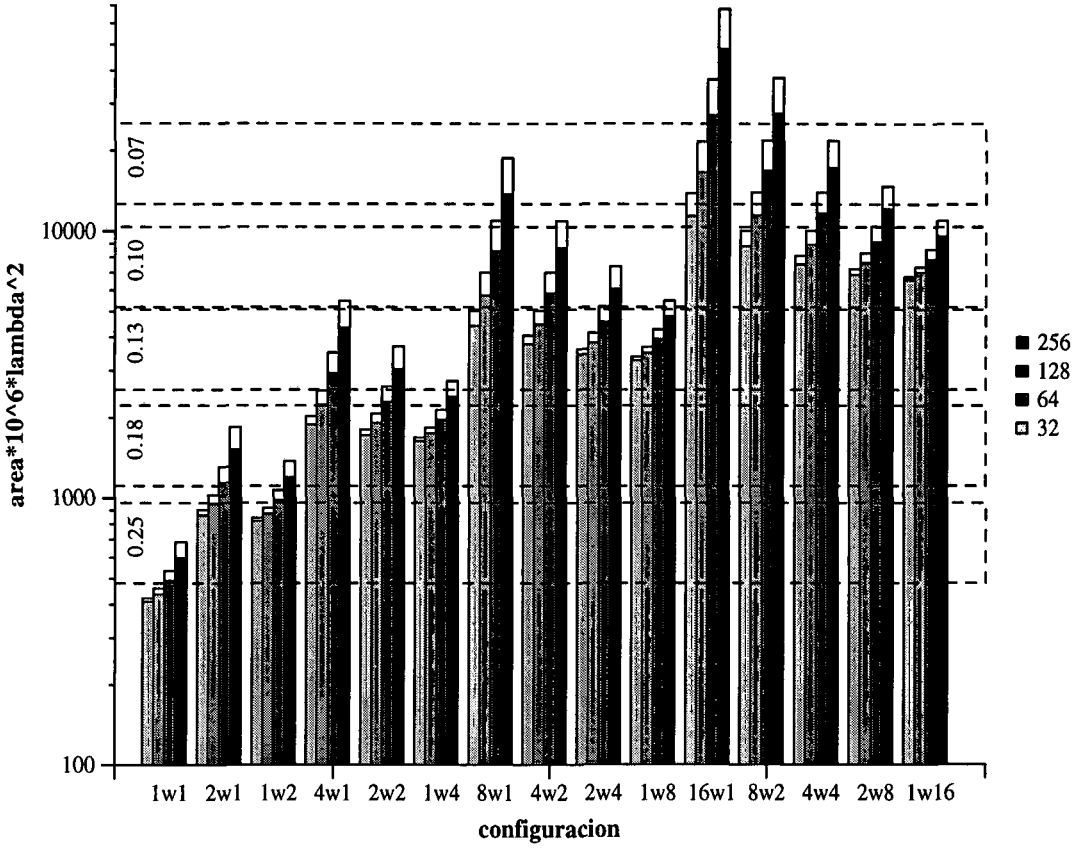

Figura 7.1: Comparativa de coste en área. La parte blanca de las barras marca el incremento de coste debido a la capacidad de implementar operaciones FMA

La tabla 7.1 muestra las configuraciones implementables para cada una de las generaciones tecnológicas consideradas. Cada generación puede implementar lo marcado en las tablas para dicha generación, más las configuraciones implementables por las generaciones anteriores. Se han considerado desde nuestra configuración base 1w1 hasta las configuraciones que multiplican por 16 su capacidad de instrucciones lanzadas por ciclo. Los bancos pueden ser de 32, 64, 128 y 256 registros, con todos los particionados posibles (que dependen de la configuración).

Configu-	32 registros					64 registros					128 registros					256 registros				
ración	1	2	4	8	16	1	2	4	8	16	1	2	4	8	16	1	2	4	8	16
1w1	✓					✓					✓					✓				
2w1	✓	✓				✓	✓				☆	☆				☆	☆			
1w2	✓					✓					☆					☆				
4w1	☆	☆	☆			☆	❑	❑			❑	❑	❑			❑	❑	✳		
2w2	☆	☆				☆	☆				❑	❑				❑	❑			
1w4	☆					☆					☆					❑				
8w1	❑	❑	❑	✳		✳	✳	✳	✳		✳	✳	●	●		●	●	●	●	
4w2	❑	❑	❑			❑	❑	✳			✳	✳	✳			✳	✳	●		
2w4	❑	❑				❑	❑				❑	❑				✳	✳			
1w8	❑					❑					❑					❑				
16w1	●	●	●	●	×	●	●	●	×	×	×	×	×	×	×	×	×	×	×	×
8w2	✳	✳	✳	●		●	●	●	●		●	●	●	×		×	×	×	×	
4w4	✳	✳	✳			✳	✳	✳			●	●	●			●	●	●		
2w8	✳	✳				✳	✳				✳	✳				●	●			
1w16	✳					✳					✳					✳				

Tabla 7.1

Configu-	32 registros					64 registros					128 registros					256 registros				
ración	1	2	4	8	16	1	2	4	8	16	1	2	4	8	16	1	2	4	8	16
1w1	✓					✓					✓					✓				
2w1	✓	✓				☆	☆				☆	☆				☆	☆			
1w2	✓					✓					☆					☆				
4w1	☆	☆	☆			❑	❑	❑			❑	❑	❑			✳	✳	✳		
2w2	☆	☆				☆	☆				❑	❑				❑	❑			
1w4	☆					☆					☆					❑				
8w1	❑	❑	✳	✳		✳	✳	✳	✳		●	●	●	●		●	●	●	×	
4w2	❑	❑	❑			✳	✳	✳			✳	✳	✳			●	●	●		
2w4	❑	❑				❑	❑				✳	✳				✳	✳			
1w8	❑					❑					❑					✳				
16w1	●	●	●	●	×	●	●	●	×	×	×	×	×	×	×	×	×	×	×	×
8w2	✳	✳	●	●		●	●	●	●		●	●	●	×		×	×	×	×	
4w4	✳	✳	✳			✳	✳	●			●	●	●			●	●	×		
2w8	✳	✳				✳	✳				●	●				●	●			
1w16	✳					✳					✳					●				

Tabla 7.2: Configuraciones implementables con una tecnología de 0.25 μ (✓), 0.18 μ (☆), 0.13 μ (❑), 0.10 μ (✳) y 0.07 μ (●). Una celda en blanco significa que esta combinación configuración / banco de registros no es posible. El símbolo × significa que esta configuración no es implementable con ninguna de las tecnologías consideradas. Los datos son para FPUs sin FMA (tabla 7.1) y con FMA (tabla 7.2)

A su vez, la tabla 7.2 muestra qué configuraciones son implementables cuando consideramos configuraciones con unidades funcionales que implementan la operación FMA (configuraciones que son más costosas en área).

Algunas configuraciones saltan de generación cuando usamos este tipo de unidades funcionales: las configuraciones 2w1(64:1) y 2w1(64:2) pasan de ser implementables con 0.25µ a necesitar una tecnología de 0.18µ; la configuración 4w1(64:1) pasa de 0.18µ a 0.13µ; las configuraciones 4w1(256:1), 4w1(256:2), 8w1(32:4), 4w2(64:10), 4w2(64:2), 2w4(128:1), 2w4(128:2) y 8w1(256:1) pasan de 0.13µ a 0.10µ; las configuraciones 8w1(128:1), 8w1(128:2), 4w2(256:1), 4w2(256:2), 8w2(32:4), 4w4(64:4), 2w8(128:1), 2w8(128:2) y 1w16(256:1) pasan de 0.10µ a 0.07µ y finalmente, la configuración 4w4(256:4) pasa de 0.07µ a no implementable con ninguna de las tecnologías estudiadas.

7.3 Adaptando la latencia de las unidades funcionales.

El tiempo descubrirá la verdad.

Séneca

Cada FPU necesita un tiempo para resolver una operación en coma flotante, tiempo que contabilizamos entre el instante en que están los datos en la entrada de la FPU hasta el instante en que ésta deja el resultado en la salida. Este tiempo es independiente del tiempo de acceso al banco de registros, es decir, si complicamos el banco de registros de manera que sea más lento su acceso, esto afectará el tiempo de ciclo del procesador, pero no hará que se tarde más o menos nanosegundos en resolver una operación.

Supongamos que tenemos un procesador con un tiempo de ciclo de X nanosegundos y que las FPUs del procesador necesitan 4 ciclos para resolver una suma en coma flotante. Si la complejidad del banco de registros nos obliga a tener un tiempo de ciclo un 33% mayor, el tiempo de resolución de la operación apenas variará, de manera que podríamos considerar que la suma se puede resolver en 3 ciclos, en lugar de los 4 ciclos anteriores (figura 7.2).

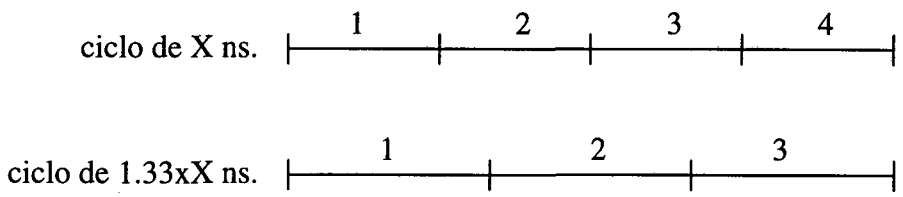

Figura 7.2: Comparativa de los ciclos necesarios para realizar una operación; 4 ciclos con un tiempo de ciclo de X ns. y 3 ciclos para un tiempo de ciclo un 33% mayor

Así pues decidimos adaptar la latencia de las unidades funcionales al tiempo de ciclo resultante de la configuración. Hemos considerado cuatro modelos de tiempo de ciclo, que se pueden ver en la tabla 7.3.

Asimismo, hemos considerado nuestra configuración base como la 1w1(32:1) y hemos asumido que seguía el modelo 4-ciclos. Cada configuración evaluada ha sido clasificada dentro de uno de los modelos antes citados según el siguiente criterio: una configuración con un tiempo de ciclo Tc relativo a la configuración base, pertenece al modelo de z-ciclos donde $z=\lceil 4/Tc \rceil$.

modelo	ciclos de las operaciones			
	store	+,*, load	div	sqrt
4-ciclos	1	4	19	27
3-ciclos	1	3	15	21
2-ciclos	1	2	10	14
1-ciclo	1	1	5	7

Tabla 7.3: Ciclos por operación de los cuatro modelos propuestos. Todas las operaciones están segmentadas, excepto div y sqrt

Configuración	Particio-nado	Banco de registros			
		32	64	128	256
1w1	1	4	4	4	3
2w1	1	3	3	3	3 / 2
	2	4 / 3	4 / 3	3	3
1w2	1	4	4	4 / 3	3
4w1	1	2	2	2	2
	2	2	2	2	2
	4	3	3	2	2
2w2	1	3	3	3 / 2	2
	2	3	3	3	3
1w4	1	4 / 3	4 / 3	3	3
8w1	1	1	1	1	1
	2	2	2	2 / 1	2 / 1
	4	2	2	2	2
	8	2	2	2	2
4w2	1	2	2	2	2
	2	2	2	2	2
	4	2	2	2	2
2w4	1	3 / 2	3 / 2	2	2
	2	3	3	3	3 / 2
1w8	1	3	3	3	3
16w1	1	1	1	1	1
	2	1	1	1	1
	4	1	1	1	1
	8	1	1	1	1
	16	1	1	1	1
8w2	1	1	1	1	1
	2	2 / 1	1	1	1
	4	2	2	2	2 / 1
	8	2	2	2	2
4w4	1	2	2	2	2 / 1
	2	2	2	2	2
	4	2	2	2	2
2w8	1	2	2	2	2
	2	3	3	2	2
1w16	1	3	3	3 / 2	2

Tabla 7.4: Modelo de ciclo para cada configuración evaluada. Cada celda marca el modelo sin FMA / con FMA. Si sólo hay un número, es que en ambos casos el modelo es el mismo

Por ejemplo, la configuración 2w4(32:1) tiene un tiempo de ciclo relativo a la base de 1.85, por lo que $\lceil 4/1.85 \rceil = \lceil 2.0513 \rceil = 3$, así que pertenece al modelo 3-ciclos. El mismo grado de replicación y *widening*, pero con un banco de 128 registros (es decir, 2w4(128:1)) tiene un tiempo de ciclo relativo de 2.09, por lo que $\lceil 4/2.09 \rceil = \lceil 1.9139 \rceil = 2$, así que pertenece al modelo 2-ciclos. Si efectuamos un particionado del banco de registros (configuración 2w4(128:2)) reducimos el tiempo de ciclo a 1.80, con lo que la configuración pertenecería al modelo 3-ciclos ($\lceil 4/1.80 \rceil = \lceil 2.22 \rceil = 3$).

En las pruebas realizadas se ha considerado el tiempo de ciclo relativo a la configuración base para adaptar las latencias de las unidades funcionales de cada configuración, con lo que logramos un modelo más ajustado a la realidad (tabla 7.4 donde se muestra el modelo para configuraciones sin FMA / con FMA). Se ha utilizado un modelo de tiempo fijo basado en parámetros de una tecnología de 0.25 λ. Los cambios en el tiempo de ciclo son debidos a modificaciones en la arquitectura. Si la tecnología nos permitiera reducir el tiempo de ciclo por un factor determinado, los resultados deberían escalarse por dicho factor.

7.4 Efectos observados

Eppur si muove

Galileo Galilei, murmullo sotto voce ante un tribunal de la inquisición

En este punto estudiaremos los efectos observados en el comportamiento de las configuraciones cuando se varía el número de registros, el grado de replicación, el grado de *widening* y el particionado del banco de registros. Estos efectos permiten comprender los resultados finales. En todos los ejemplos que veremos a continuación se han considerado configuraciones con FPUs que no implementan la operación FMA.

La figura 7.3 muestra el efecto de incrementar el banco de registros para una configuración fija (la configuración 1w1, con banco de registros sin particionar). Se puede observar cómo al aumentar el número de registros de 32 a 64 se produce una mejora de rendimiento (a costa de un incremento en el área requerida), mientras que al pasar de 64 a 128

o a 256 se produce una pérdida de rendimiento. Ello es debido a que ésta es una configuración poco agresiva, donde con un banco de 64 registros casi no se produce *spill code*. Así, un banco de más de 64 registros tiene un incremento de rendimiento muy pequeño y un incremento de tiempo de ciclo bastante mayor, por lo que no vale la pena utilizar bancos con muchos registros. La conclusión de este efecto es que cuando se tiene un banco de registros suficientemente grande como para necesitar poco *spill code*, cualquier incremento del banco produce una pérdida de rendimiento. Este efecto beneficia a la técnica de *widening*, ya que, al tener registros anchos, se puede eliminar el *spill code* con un banco de registros menor que el necesario cuando se aplica la técnica de replicación.

Las figuras 7.4a y 7.4b muestran el efecto de aplicar las técnicas de replicación y *widening*, respectivamente, con un banco de 128 registros. En todos los casos se ha considerado la mayor partición posible del banco, que es la que ofrece un tiempo de ciclo menor. En la figura 7.4a se puede observar cómo conforme se incrementa el grado de replicación, el incremento de rendimiento se reduce, hasta convertirse en pérdida de rendimiento (en la transición de 4w1(128:4) a 8w1 (128:8)). Esto es debido a dos razones:

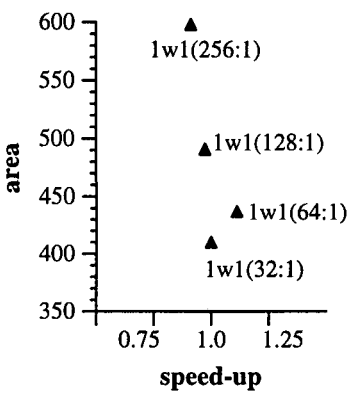

Figura 7.3: Efecto producido por el incremento del número de registros

- bucles que alcanzan el rendimiento máximo. Conforme se aumenta el grado de replicación, cada vez más bucles han alcanzado su grado de paralelismo máximo, con lo que el incremento de instrucciones ejecutadas por ciclo (IPC) es cada vez menor. Así, el incremento de IPC es contrarrestado por el enorme aumento de tiempo de ciclo debido al número de puertos añadido al banco de registros.

- el incremento en la presión sobre los registros que produce el hecho de pasar a configuraciones cada vez más agresivas. Esta presión puede provocar un incremento en las necesidades de *spill code* que redunda en una pérdida de rendimiento.

Además, las configuraciones con alto grado de replicación pueden ser inimplementables, pues pueden llegar a ocupar más del 20% del área del *chip*, que es el límite que nos marcamos.

La figura 7.4b muestra el efecto de aplicar *widening*. Aumentando el grado de *widening* se produce un incremento en área y tiempo de ciclo menor que al aumentar el grado de replicación. El menor incremento en área de *widening* puede permitir tener un banco de

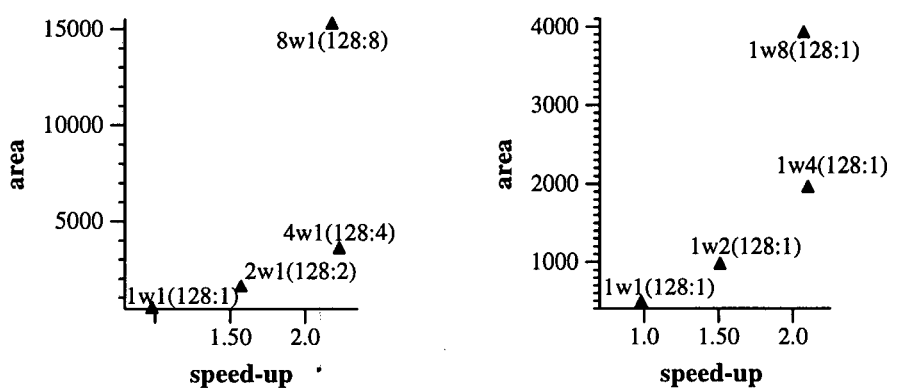

Figura 7.4: Efecto de aplicar en solitario las técnicas de replicación (a) y widening (b)

registros mayor sin alcanzar el límite del 20% del total del área del *chip* (nótese que las figuras 7.4a y 7.4b no están a la misma escala). Además, el incremento de la capacidad de almacenamiento de la técnica de *widening* producida por tener registros anchos, junto con un tiempo de ciclo menor que el de la técnica de replicación resulta en un buen rendimiento. A pesar de ello, hay un punto en que el incremento de IPC es lo suficientemente pequeño como para que el rendimiento final empiece a degradarse.

La figura 7.5 muestra el efecto de diversas configuraciones con ancho de banda 8 (es decir, configuraciones 8w1, 4w2, 2w4 y 1w8). El banco de registros ha sido fijado a 128 entradas, con el mayor particionado posible para minimizar el tiempo de ciclo. Podemos observar el desigual incremento en área al aplicar replicación y *widening*, que puede forzar que una configuración no sea implementable: en el ejemplo las configuraciones 1w8(128:1) y 2w4(128:2) son implementables con una tecnología de λ=0.13, 4w2(128:4) con una tecnología de λ=0.10 y 8w1(128:8) con una de λ=0.07. Asimismo podemos observar cómo la combinación de replicación y *widening* nos ofrece el mejor rendimiento.

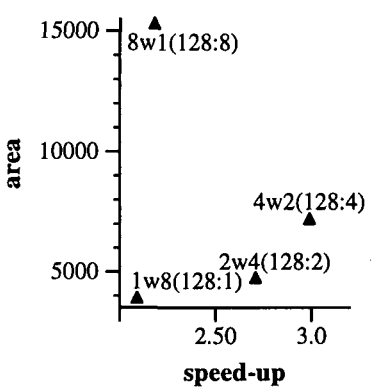

Figura 7.5: Efecto de configuraciones con un rendimiento teórico igual a 8 veces el de la configuración base

7.5 Configuraciones con mejor rendimiento por generación tecnológica.

Esto es todo, amigos.

Porky Pig, en los dibujos de la Warner

Como hemos dicho, se han probado las configuraciones implementables por coste en área para cada una de las cinco generaciones tecnológicas propuestas por la SIA (λ=0.25, 0.18, 0.13, 0.10 y 0.07). Las configuraciones varían el grado de replicación y *widening* (1, 2, 4, 8 y 16) el número de registros del banco (32, 64, 128 y 256) así como su particionado. Dado que el resultado de estas variaciones son 140 configuraciones (280 si consideramos FPUs capaces de implementar la operación FMA), presentamos sólo las cinco configuraciones con mejor rendimiento para cada tecnología. La figura 7.6 muestra las cinco mejores entre las configuraciones con FPUs que no implementan FMA, y la figura 7.7 muestra las cinco mejores entre todas las configuraciones.

Aunque se presenten las cinco configuraciones con mejor rendimiento para cada generación tecnológica, no todas ellas son las que ofrecen mejor relación rendimiento/coste. Consideraremos una configuración "elegible" si todas las configuraciones que alcanzan su rendimiento tienen un coste superior. Así, una configuración será no "elegible" si existe otra configuración con menos coste y más rendimiento. Para cada gráfica se muestran en negro las configuraciones "elegibles" y en gris las no "elegibles". Comentaremos primero los resultados de la figura 7.6.

La mayoría de las configuraciones presentadas en la figura 7.7 aplican la técnica de *widening*, pero cuando miramos sólo las elegibles, todas utilizan dicha técnica. El efecto es debido a la combinación de mayor capacidad de almacenamiento al tener los registros anchos junto con la reducción del tiempo de ciclo del banco de registros. Por ejemplo, para una tecnología de 0.25μ (figura 7.7a), la configuración 1w2(64:1) tiene un incremento de tiempo de ciclo relativo de 1.15 y puede almacenar 128 palabras, mientras que 2w1(64:1) tiene un

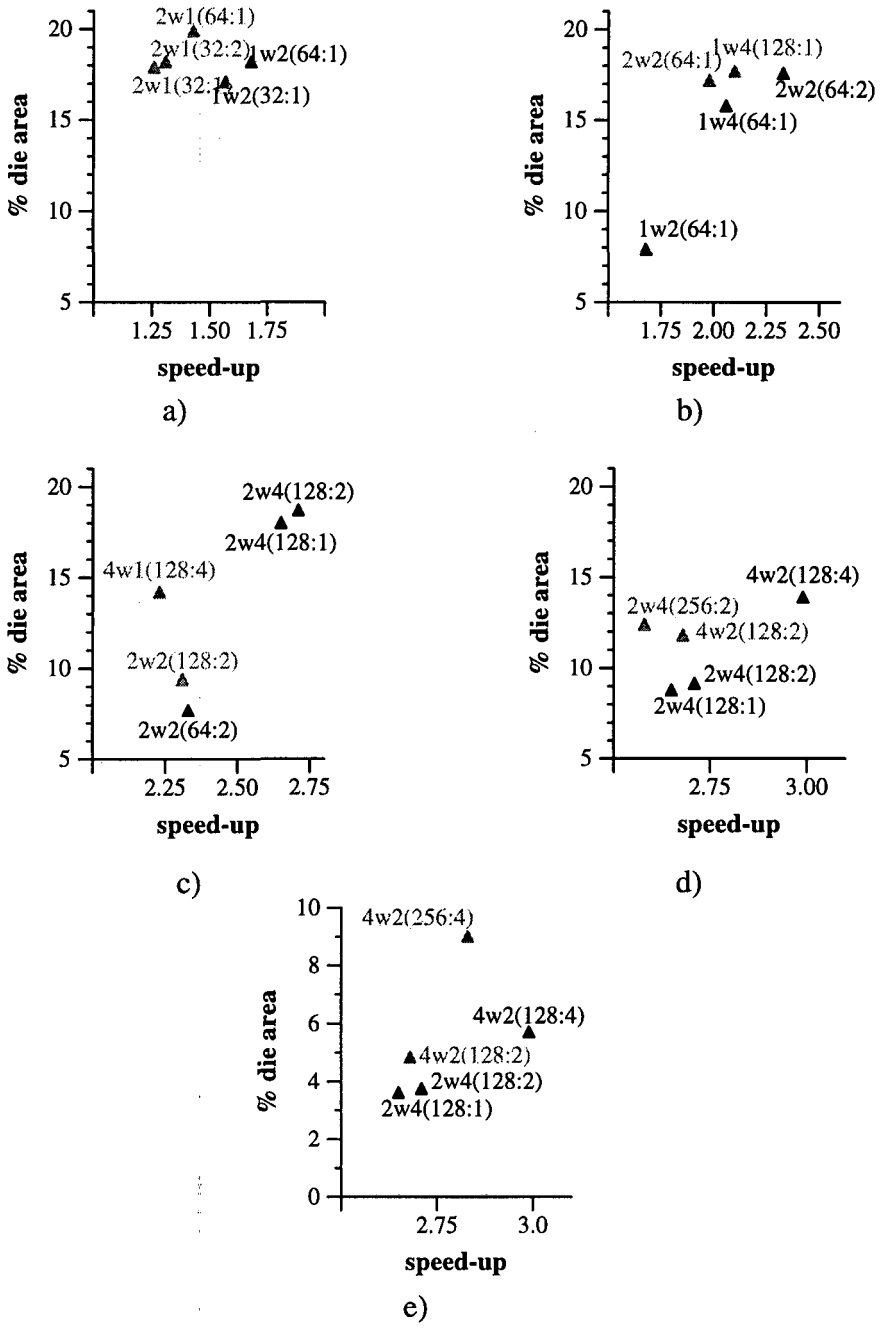

Figura 7.6: Las cinco configuraciones (entre las que no implementan FMA) que ofrecen mejor rendimiento para cada una de las generaciones tecnológicas propuestas por la SIA: a) 0.25 b) 0.18, c) 0.13 d) 0.10 y e) 0.07

tiempo relativo de 1.54 y almacena sólo 64 palabras. Este efecto puede observarse en las cinco generaciones tecnológicas.

Podemos observar cómo configuraciones con un rendimiento pico 16 veces superior respecto a la base (1w16, 2w8, 4w4, 8w2 y 16w1) no aparecen en las gráficas a pesar de que algunas se pueden implementar con una tecnología de 0.10 μ y la mayoría con tecnología de 0.07 μ. Esto es debido a dos razones: cuando dichas configuraciones tienen un alto grado de *widening*, existen configuraciones con menor rendimiento pico, pero con alto grado de replicación, que tienen mejor rendimiento (por ejemplo 4w2 tiene mejor rendimiento que 2w8). Por otro lado, las configuraciones con rendimiento pico 16 veces superior a la base y con un alto grado de replicación, tienen un coste prohibitivo en área y, sobre todo, en tiempo de ciclo.

Cabe destacar que cuatro de las cinco mejores configuraciones bajo una tecnología de 0.10μ también lo son bajo una tecnología de 0.07μ. Únicamente la configuración 4w2(256:4) desplaza a la 2w4(256:2) al pasar de 0.10 a 0.07, debido a que 4w2(256:4) no era implementable con la primera tecnología (pero ambas son no "elegibles"). Un efecto interesante es que ninguna de las cinco configuraciones que ofrecen mejor rendimiento con λ=0.07 supera el 10% de área del chip. Además, la configuración con mejor rendimiento, 4w2(128:4) ocupa el 6.01% del área total. De ello se desprenden dos conclusiones. En primer lugar, el hecho de que empleando replicación y *widening* conjuntamente se puede conseguir el mejor rendimiento reduciendo el coste (como consecuencia se puede permitir mejorar otros aspectos del procesador, como tener *on-chip caches* mayores) y en segundo lugar, habrá que tener en cuenta que a partir de una determinada generación tecnológica, aumentar el número de instrucciones por ciclo requerirá nuevas técnicas, diferentes de las estudiadas.

Los puntos anteriormente expuestos son válidos asimismo para las gráficas de la figura 7.7, por lo que a la hora de analizar dichas gráficas nos limitaremos a comentar las diferencias entre usar FPUs con FMA y sin FMA. En la figura 7.7, las configuraciones que implementan

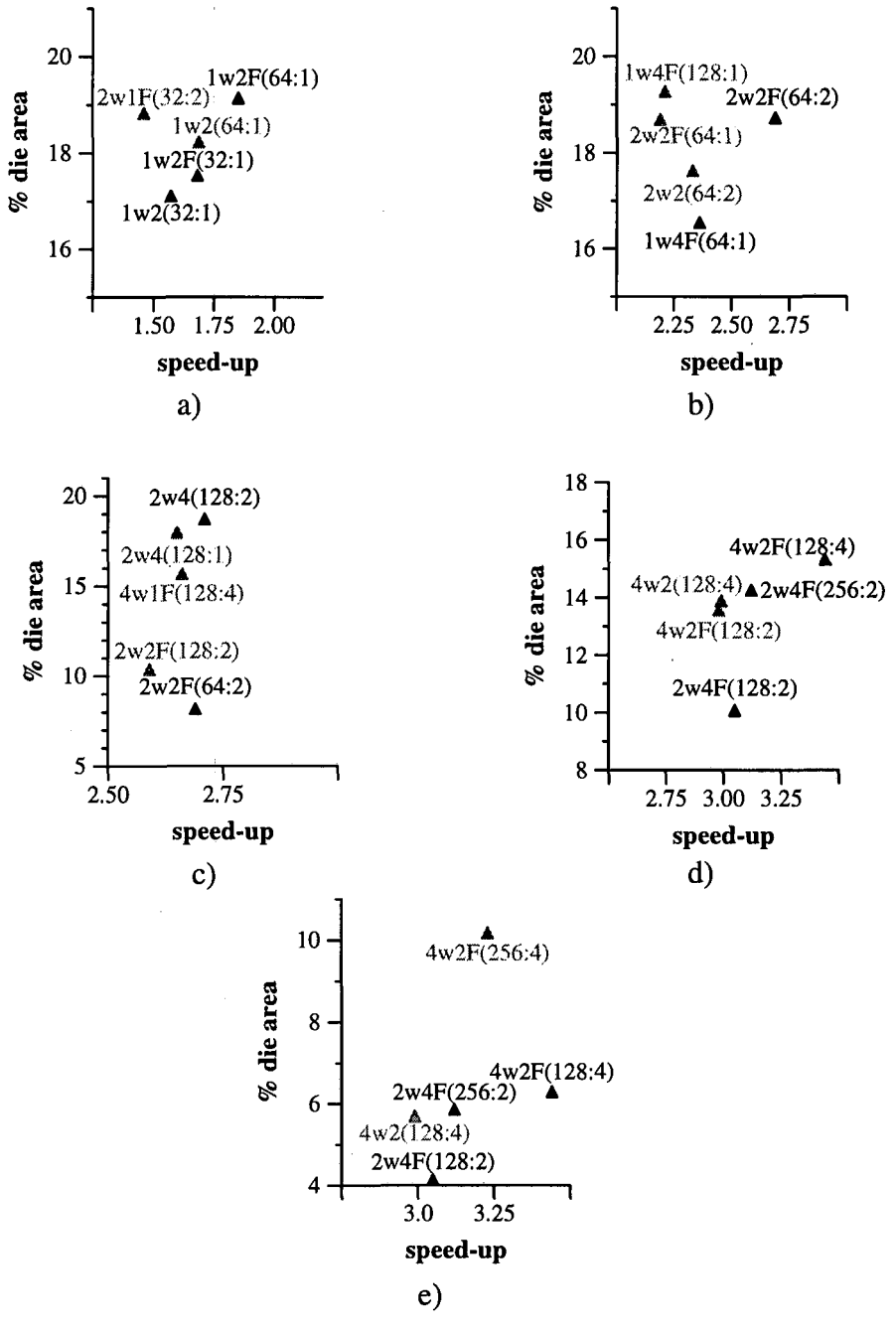

Figura 7.7: Las cinco configuraciones que ofrecen mejor rendimiento para cada una de las generaciones tecnológicas propuestas por la SIA: a) 0.25 b) 0.18, c) 0.13 d) 0.10 y e) 0.07

FPUs con FMAs siguen la notación XwY**F**(Z:n), mientras que las que no la implementan siguen la notación XwY(Z:n).

Una de las primeras observaciones que se pueden extraer de las gráficas de la figura 7.7 es que hay más configuraciones con FMA que sin FMA entre las cinco mejores de cada tecnología. En la tecnología de 0.25μ, la relación es de 3 a 2 a favor de las que implementan FMA. Esta relación es de 4 a 1 con 0.18μ, de 3 a 2 con 0.13μ y de 4 a 1 con 0.10μ y 0.07μ. Si miramos pues las configuraciones con mejor relación rendimiento-coste entre las cinco mejores de cada tecnología, la balanza se decanta todavía más hacia las configuraciones con FMA: de las 25 configuraciones mostradas, 13 son las "elegibles", y de estas, 11 usan unidades funcionales que implementan FMA.

De estos números se puede concluir que el incremento de rendimiento en ciclos debido a usar FPUs con FMA es lo suficientemente grande, en general, como para no quedar contrarrestado por el incremento en tiempo de ciclo debido al uso de este tipo de FPUs. Hemos de considerar que dadas dos configuraciones cuya única diferencia sea el tipo de FPUs, el incremento de rendimiento en ciclos entre la que implementa la operación FMA y la que no la implementa es debido a tres factores (como vimos en el capítulo 5):

- el rendimiento teórico: el *MII* de una configuración con FMA suele ser menor.
- el factor *spill code*: una configuración con FMA tiene una operación producto y una operación suma asociada fusionadas, con lo que el resultado intermedio entre el producto y la suma no es necesario guardarlo en ningún registro. Esto significa una reducción en los requerimientos de registros que puede resultar en una necesidad menor de *spill code* (y un mayor rendimiento).
- el factor planificador: el planificador es una heurística que no siempre consigue alcanzar el *MII*. La cantidad de ciclos añadida por el planificador suele ser mayor cuanto más complejo sea el bucle a planificar. Si fusionamos varias operaciones, el número de nodos a planificar se reduce, dando más oportunidades al planificador.

7.6 Sumario y contribuciones

En este capítulo se ha presentado un estudio de rendimiento de las técnicas descritas en los capítulos anteriores (replicación, *widening* y fusión de multiplicación y suma), teniendo en cuenta su coste en área y tiempo de ciclo. La metodología utilizada para la realización de dicho estudio consiste en:

- calcular qué configuraciones tienen un coste en área que no exceda el máximo permitido para cada generación tecnológica (según las previsiones de la SIA).
- calcular el tiempo de ciclo de cada configuración.
- adaptar la latencia de las unidades funcionales de cada configuración dependiendo del tiempo de ciclo de la misma.
- calcular el rendimiento en ciclos de cada configuración para bancos de 32, 64, 128 y 256 registros con todas las particiones posibles.
- calcular el rendimiento de las configuraciones en función de su rendimiento en ciclos y del tiempo de ciclo requerido.

De este estudio podemos concluir que la combinación de pequeños grados de replicación y *widening* junto con el fusionado de FMAs, ofrece el mejor rendimiento en la parte en coma flotante de un procesador (para código numérico) cuando se tiene en cuenta factores como el *spill code* y los costes (área ocupada y tiempo de ciclo). Los factores que producen este efecto son:

- la combinación de las técnicas de replicación y *widening* ofrece un rendimiento teórico cercano a emplear únicamente replicación.
- el aumento de la capacidad de almacenamiento en la técnica de *widening* provoca un incremento de rendimiento cuando se tiene en cuenta el *spill code.*

- replicación hace crecer el coste en área cuadráticamente, mientras que *widening* lo hace linealmente.

- el tiempo de ciclo del banco de registros empleando replicación es mucho mayor que aplicando *widening*, por lo que la combinación de ambas ofrece un buen rendimiento en ciclos, con un tiempo de ciclo relativamente pequeño.

- la fusión de multiplicación y suma tiene un incremento en rendimiento que no queda contrarrestado por la penalización en tiempo de ciclo debida al uso de este tipo de FPUs.

La aportación del capítulo es el estudio completo, realizado, que tiene en cuenta el rendimiento teórico, las limitaciones impuestas por el planificador y el banco de registros, y los costes de cada técnica evaluada.