



ESCOLA TÈCNICA SUPERIOR
D'ENGINYERIA
Universitat Rovira i Virgili



Arquitectura de Computadors

Pràctica 1

curs 2024-25

Estudiant: Gemma Goitia Gál i Ivan
Garcia Pallares

Grup Laboratori: LX

Professor/a: Carles Aliagues

Data de lliurament: 09/10/2024

Índex

Arquitectura de Computadors

Pràctica 1

curs 2024-25

1

Índex

2

Característiques de Disseny dels Processadors

3

Simulacions Realitzades

9

Estudi de Millores

14

Comentari del Paràgraf

22

WEBGRAFIA

23

Característiques de Disseny dels Processadors

Paràmetres a cercar (i les seves descripcions):

- 1) *La k-via del processador → quantitat d'instruccions per cicle que poden arribar a tractar a l'hora.*
- 2) *La mida dels buffers que emmagatzemen instruccions.*
- 3) *Les caches L1 i L2. Si manipulen per separat instruccions i dades i la mida, associativitat i algorisme de reemplaçament.*
- 4) *L'amplada de banda i latència de la memòria principal.*

L'amplada de banda fa referència al número de línees de dades que connecta el processador amb la memòria RAM.

La latència és el temps que tarda en obtenir dades des de la memòria RAM.

- 5) *Unitats Funcionals. Numeros d'ALU's aritmètiques i multiplicació d'entegers, ALu's aritmètiques i multiplicació de coma flotant i el nombre de ports d'accés a memòria de primer nivell de cachés.*

Intel Core i5-14400 (Lackluster core)

El Intel Core i5-14400 utilitza la arquitectura "Raptor Lake Refresh"

K-via (IPC) $\Rightarrow 4$

Mida dels buffers \Rightarrow

Finestra instruccions (ruu) $\Rightarrow 4\text{Kb}$

Cua d'accés a memòria (lsq) \Rightarrow load: 144 bytes, store: 192 bytes

Cachés \Rightarrow

| | |
|------------|--------------------|
| Cache L1: | 80 KB (per core) |
| Cache L2: | 1.25 MB (per core) |
| Cache L3: | 20 MB (compartida) |
| E-Core L1: | 96 KB (per core) |
| E-Core L2: | 2 MB (per module) |

Associativitat : associativa per conjunts.

Algorisme de substitució: l'algoritme utilitzat per intel en els seus últims processadors ha variat depenent del nivell de la caché. LRU (Last Recently Used), FIFO (First In, First Out) i un que combina ambdues utilitzant cada un depenent de la taxa d'encerts, la latència i la capacitat de hardware, entre altres.

Amplada de banda Memòria Principal (quantitat de dades que poden transferir-se per unitat de temps) $\Rightarrow 4$ instruccions per ciclo

Latència Memòria Principal (temps que tarda en accedir a una dada específica) $\Rightarrow 5$ ciclos

Unitats Funcionals \Rightarrow

ALUs integers $\Rightarrow 5$

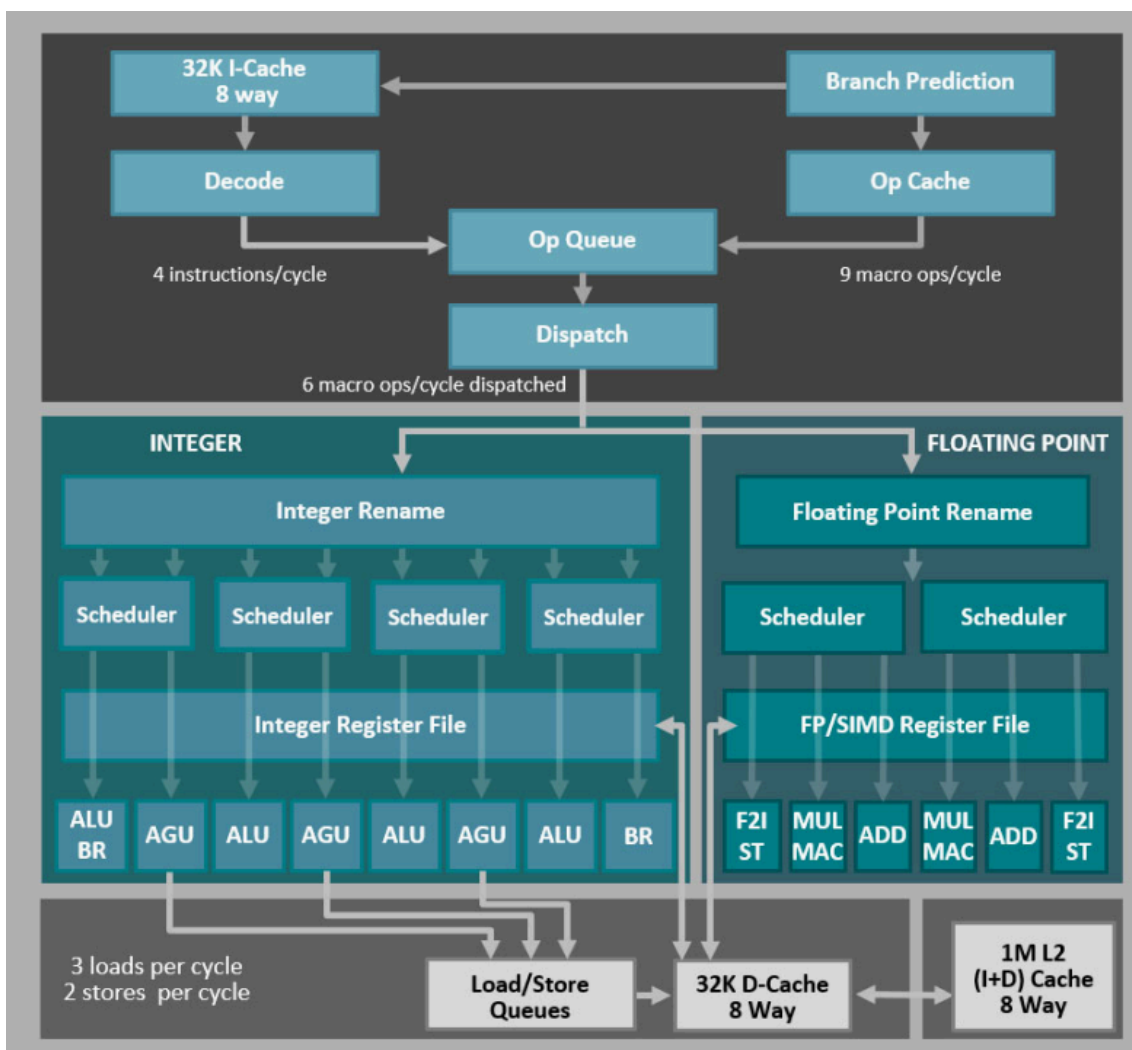
ALUs coma flotant $\Rightarrow 4$

Núm. ports d'accés a memòria de primer nivell de cachés $\Rightarrow 7$

AMD Ryzen 5 7600X (Zen 4 core)

El procesador AMD Ryzen 5 7600X utilitza l'arquitectura "Zen4".

Aquesta arquitectura utilitza cinc caches en una jerarquia de 3 nivells per accelerar l'execució d'instruccions i el processament de dades. En el primer nivell trobem la caché de dades, la d'instruccions i la de OP. Seguidament, en la segona trobem una per core de dades i instruccions juntes i finalment al tercer nivell trobem una última de fins a 32 Mbytes.





K-via (IPC) ⇒

Fetch ⇒ 6 Decode ⇒ 4 Issue ⇒ 14 Commit ⇒ 8

Mida dels buffers ⇒

Finestra instruccions (ruu) ⇒ 320 bytes (re-order buffer)

Cua d'accés a memòria (lsq) ⇒ 152 bytes

Cachés ⇒

Cache L1 d'Instruccions: té un tamany de 32 Kbytes amb una associativitat de 8. Les línies de la caché (mida de bloc) són de 32 bytes. Les funcions associades amb aquesta caché L1 són obtenir línies de memòria de la L2, proporcionar bytes d'instruccions al decodificar i precarregar les instruccions. Si allí no hi són passarà a fer la recerca a la L3. Les sol·licituds que manquen en tots els nivells s'obtenen de la memòria del sistema. L'algoritme de substitució correspon al LRU.

Cache L1 de Dades: té una mida de 32 Kbytes amb una associativitat de 8 i una mida de bloc 32 bytes. És una caché write-back que admet fins a 3 operacions per cicle. L'algoritme de substitució també és LRU.

Cache L2 de Dades i Instruccions: tamany de 1Mbyte amb una latència variable de càrrega a ús de no menys de 14 cicles. Té una associativitat de 8, una mida de bloc 32 bytes i l'algoritme és LRU.

Caché L3: pot arribar als 32 Mbytes amb una associativitat de 16.

Caché OP: té 64 sets i una associativitat de 12 vies. conté instruccions prèviament descodificadas. Quan s'obtenen les instruccions d'aquí, no s'ha de fer la recuperació i descodificació d'aquestes, cosa que millora la latència del pipeline perquè és més curt que el pipeline tradicional de recuperació i descodificació.

Amplada de banda i latència de la memòria principal ⇒

L'amplada de banda es calcula multiplicant la freqüència de treball del bus, en cicles per segon per el número de bytes que es transfereixen en cada cicle.

Amplada Banda (bytes / cicle) = Max Memory BW / Freqüència

(83.2 GB/s) / (4.7 GHz) = 17,702127 (bytes / cicle)

Ho arrodonirem a 32 bytes / cicle a la simulació.

Sabent que tenim les dades de:

Freqüència del processador ⇒ 4,7 GHz (encara que pot arribar a 5,3 GHz)

Freqüència de la memòria ⇒ 400 MHz

CAS de memòria ⇒ 12,5 ns

tRCD de memòria ⇒ 20

Taxa de dades de memòria ⇒ 3200 MT/s

Hem agafat les dades de la RAM DDR4-3200W.

Per poder-ho convertit en el paràmetre `-mem:lat` pel simplescalar s'han utilitzat les següents operacions:

$$\text{first_chunk} = \frac{\text{CPU_Clock} \times (\text{CAS} + t\text{RCD})}{\text{Memory_Clock}}$$
$$\text{inter_chunk} = \frac{\text{CPU_Clock}}{\text{Memory_Data_Rate}}$$

First chunk ⇒ són els cicles de rellotge necessaris per portar el primer bloc de memòria a caché d'una petició.

$(4.7 * (12.5 + 20)) / (400) = 47,7343$

Inter chunk ⇒ són els cicles de rellotge necessaris per portar els següents blocs de memòria a caché de aquella mateixa petició.

$$(4.7) / (3200) = 1,46875$$

Unitats Funcionals ⇒

ALUs totals ⇒ 6

ALUs integers ⇒ 4

ALUs coma flotant ⇒ 2

Núm. ports d'accés a memòria de primer nivell de cachés ⇒ 2

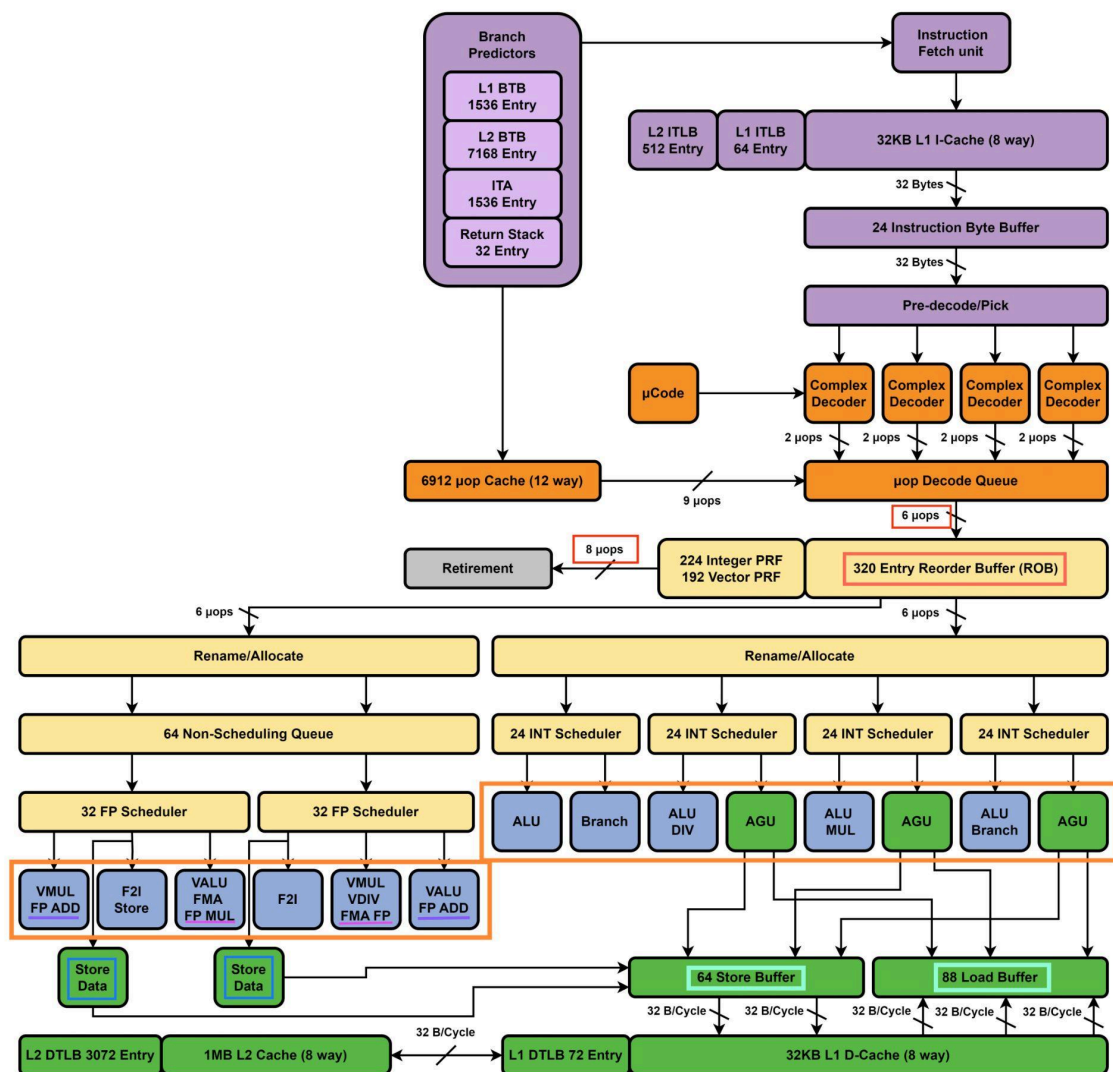
Explicació dels paràmetres trobats mitjançant el diagrama del Zen4:

AMD - 2022

Zen 4

By Cardyak

Microarchitecture Block Diagram



- Vermell: informació del commit i el decode.
- Rosa: dades finestra d'instruccions.
- Taronja: issue.
- Blau: ports d'accés a memòria.
- Fucsia: ALUs de coma flotant.
- Turquesa Clar: cua d'accés a memòria. $64 + 88 = 152$.

Simulacions Realitzades

PER CALCULAR EL NÚMERO DE SETS PER LA SIMULACIÓ DEL SIMPLESCALAR:

$$N^{\circ} \text{ de sets} = (1024 * \text{tamany total}) / (\text{associativitat} * \text{tamany de bloc})$$

Per poder realitzar la simulació, hem acabat de calcular la informació restant necessària sobre les caches de cada processador per poder executar les comandes.

Dades obtingudes de les caches del Intel Core i5-14400

| | Caché DL1 | Caché IL1 | Caché L2 |
|----------------|-----------|-----------|----------|
| Tamany Total | 32 KB | 1024 KB | 1024 KB |
| Associativitat | 8 | 8 | 8 |
| Mida de Bloc | 32 | 32 | 32 |
| Nº de Sets | 128 | 4096 | 4096 |

$$N^{\circ} \text{ de sets DL1} = (1024 * 32) / (8 * 32)$$

$$N^{\circ} \text{ de sets IL1} = (1024 * 1024) / (8 * 32)$$

$$N^{\circ} \text{ de sets L2} = (1024 * 1024) / (8 * 32)$$

Dades obtingudes de les cachés del Intel AMD Ryzen 5 7600X

| | <i>Caché DL1</i> | <i>Caché IL1</i> | <i>Caché L2</i> |
|-----------------------|------------------|------------------|-----------------|
| <i>Tamany Total</i> | <i>32 KB</i> | <i>32 KB</i> | <i>1024 KB</i> |
| <i>Associativitat</i> | <i>8</i> | <i>8</i> | <i>8</i> |
| <i>Mida de Bloc</i> | <i>32</i> | <i>32</i> | <i>32</i> |
| <i>Nº de Sets</i> | <i>128</i> | <i>128</i> | <i>4096</i> |

$$N^{\circ} \text{ de sets DL1} = (1024 * 32) / (8 * 32)$$

$$N^{\circ} \text{ de sets IL1} = (1024 * 32) / (8 * 32)$$

$$N^{\circ} \text{ de sets L2} = (1024 * 1024) / (8 * 32)$$

Comanda per la simulació del Intel Core i5-14400

Configuració:

[illegible]

He usado una ejecución usando unos los datos en un archivo aparte sin ponerlos directamente en la comanda. Y lo he usado para una ejecución genérica de este archivo para que ejecute todos los benchmarks a la vez.

Y una vez finalizados los resultados los digia a una carpeta controlada y de ahí sacamos los ipc de cada benchmark y los guardamos en un archivo con los resultados y a parte los imprimimos con un 'cat' para facilitar la faena.

Millora del fetch i el decode:

```
-fastfwd 100000000
-max:inst 100000000
-fetch:ifqsize 12
-decode:width 12
-issue:width 12
-commit:width 12
-ruu:size 256
-lsq:size 256
-mem:lat 5 2
-mem:width 4
-res:ialu 5
-res:imult 5
-res:mempport 7
-res:fpalu 4
-res:fpmult 5
-cache:dl1 dl1:128:32:8:l
-cache:il1 il1:2048:32:8:l
-cache:dl2 ul2:2048:32:8:l
```

Millora del ruu:

```
-fastfwd 100000000
-max:inst 100000000
-fetch:ifqsize 6
-decode:width 6
-issue:width 12
-commit:width 12
-ruu:size 512
-lsq:size 256
-mem:lat 5 2
-mem:width 4
-res:ialu 5
-res:imult 5
-res:mempport 7
-res:fpalu 4
-res:fpmult 5
-cache:dl1 dl1:128:32:8:l
-cache:il1 il1:2048:32:8:l
-cache:dl2 ul2:2048:32:8:l
```

Millora Caché L1:

```
-fastfwd 100000000
-max:inst 100000000
-fetch:ifqsize 6
-decode:width 6
-issue:width 12
-commit:width 12
-ruu:size 256
-lsq:size 256
-mem:lat 5 2
-mem:width 4
-res:ialu 5
-res:imult 5
-res:mempport 7
-res:fpalu 4
-res:fpmult 5
-cache:dl1 dl1:256:32:8:l
-cache:il1 il1:2048:32:8:l
-cache:dl2 ul2:2048:32:8:l
```

3 Millores:

```
-fastfwd 100000000
-max:inst 100000000
-fetch:ifqsize 12
-decode:width 12
-issue:width 12
-commit:width 12
-ruu:size 512
-lsq:size 256
-mem:lat 5 2
-mem:width 4
-res:ialu 5
-res:imult 5
-res:mempport 7
-res:fpalu 4
-res:fpmult 5
-cache:dl1 dl1:256:32:8:l
-cache:il1 il1:2048:32:8:l
-cache:dl2 ul2:2048:32:8:l
```

Amb els canvis anteriors hem obtingut els següents resultats:

| Benchmarks | ampp | applu | eon | equake | vpr |
|------------|--------|-------|--------|--------|--------|
| IPS | 0,7121 | 5,031 | 3,3744 | 3,548 | 1,9952 |

Cambio duplicando la caché L1 para ver si almacenando más información en la caché más rápida nos beneficia aunque tarde más en bucar la información

| Benchmarks | ampp | applu | eon | equake | vpr |
|------------|--------|-------|-------|--------|--------|
| IPS | 0,7122 | 5,031 | 3,347 | 3,548 | 1,9952 |

Duplicamos la ventana de instruccines a memoria (ruu)

| Benchmarks | ampp | applu | eon | equake | vpr |
|------------|-------|-------|--------|--------|--------|
| IPS | 0,905 | 5,031 | 3,3744 | 3,5881 | 1,9758 |

Duplicamos la cantidad de instrucciones que pueden entrar y descifrar

| Benchmarks | ampp | applu | eon | equake | vpr |
|------------|--------|--------|--------|--------|--------|
| IPS | 0,9149 | 6,2034 | 3,6169 | 3,6959 | 2,0662 |

Aplicamos las 3 mejoras a la vez

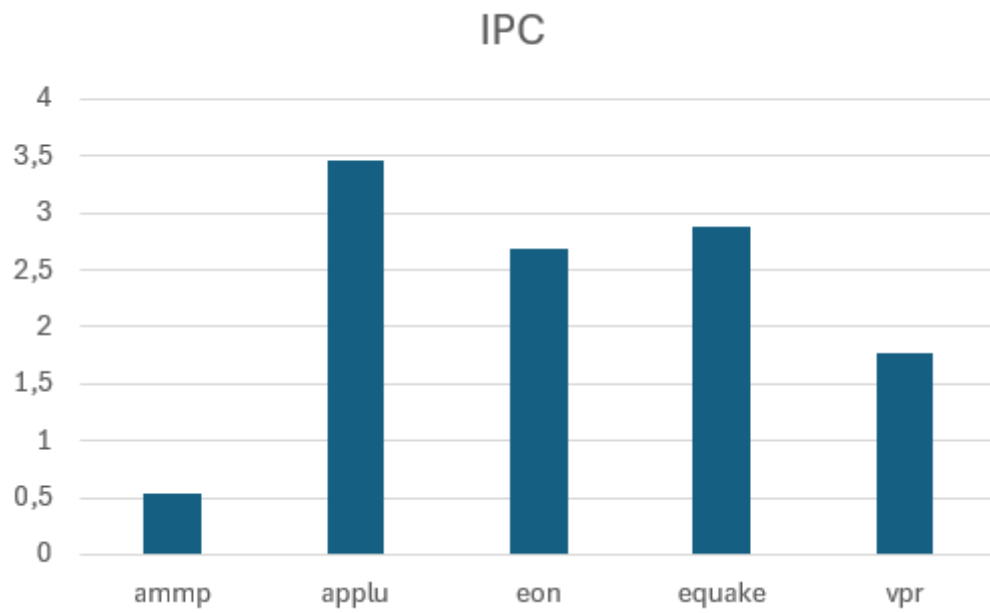
| Benchmarks | ampp | applu | eon | equake | vpr |
|------------|--------|--------|--------|--------|--------|
| IPS | 0,9149 | 6,9524 | 3,6182 | 3,6678 | 2,0723 |

Comanda per la simulació del AMD Ryzen 5 7600X

```
-cache:dl1    dl1:128:32:8:1    -cache:il1    il1:128:32:8:1    -cache:dl2  
ul2:4096:32:8:1 -fetch:ifqsize 6 -decode:width 4 -issue:width 14  
-commit:width 8 -ruu:size 512 -lsq:size 256 -mem:lat 48 1 -mem:width  
32 -res:ialu 4 -res:imult 2 -res:mempport 2 -res:fpalu 2 -res:fpmult 1
```

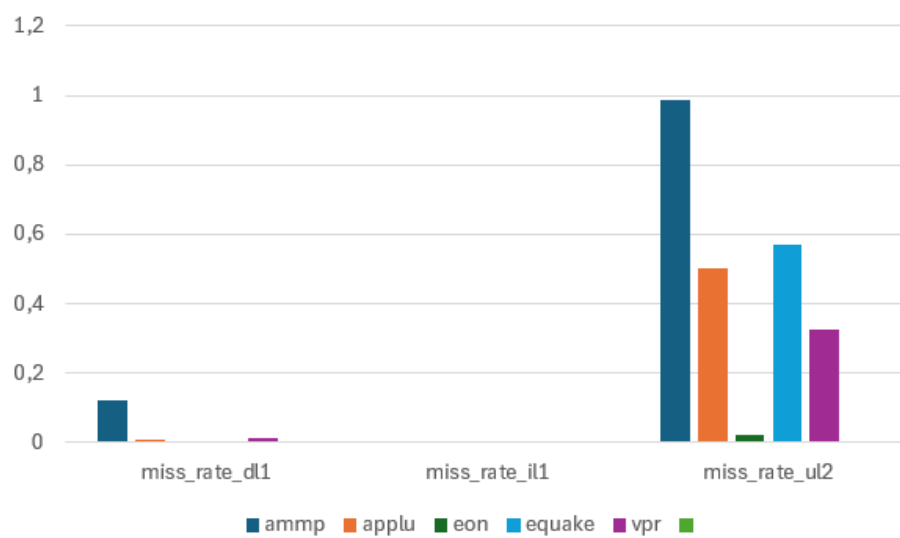
Executant les comandes amb els paràmetres indicats anteriorment hem obtingut les següents dades sobre l'IPC en els diferents benchmarks:

| <i>Benchmark</i> | <i>ammp</i> | <i>applu</i> | <i>eon</i> | <i>equake</i> | <i>vpr</i> |
|------------------|-------------|--------------|------------|---------------|------------|
| <i>IPC</i> | 0,5329 | 3,4529 | 2,6891 | 2,8828 | 1,7719 |



També hem recol·lectat les dades de miss de les 3 principals caches.

| | <i>ammp</i> | <i>applu</i> | <i>eon</i> | <i>equake</i> | <i>vpr</i> |
|----------------------|-------------|--------------|------------|---------------|------------|
| <i>miss_rate_dl1</i> | 0,1233 | 0,0068 | 0,0017 | 0,0004 | 0,0126 |
| <i>miss_rate_il1</i> | 0,000 | 0,000 | 0,000 | 0,000 | 0,000 |
| <i>miss_rate_ul2</i> | 0,9862 | 0,5017 | 0,0219 | 0,5704 | 0,3295 |



Podem observar com en les caches de primer nivell el miss rate és quasi null a diferència de la de segon nivell. Això pot ser degut a un tamany massa petit que faci que les dades s'hagin d'anar a buscar al següent nivell i es perdi rendiment.

Estudi de Millores

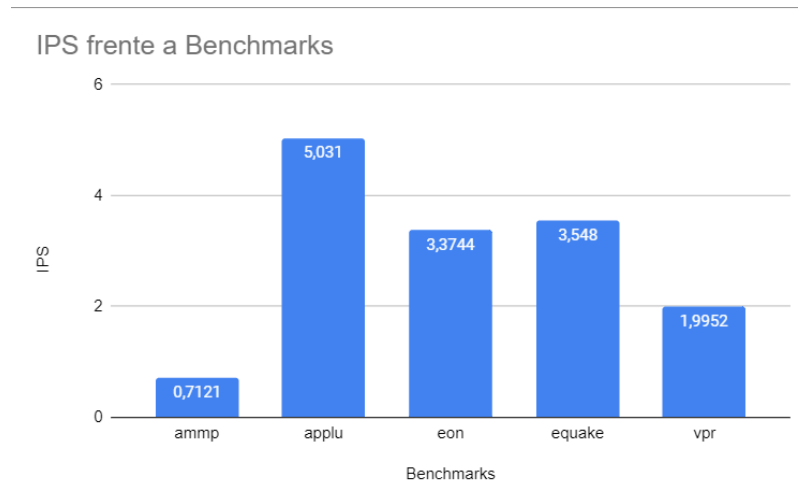
Intel i5-14400

Valors de la nostra CPU:

| Parámetros | Valores Pred | 1r Cambio | 2o Cambio | 3r Cambio | Cambios Acumulados |
|------------|--------------|-----------|-----------|-----------|--------------------|
| fetch | 6 | 6 | 6 | 12 | 12 |
| decode | 6 | 6 | 6 | 12 | 12 |
| issue | 12 | 12 | 12 | 12 | 12 |
| commit | 12 | 12 | 12 | 12 | 12 |
| ruu | 256 | 256 | 512 | 256 | 512 |
| lsq | 256 | 256 | 256 | 256 | 256 |
| latencia | 5 | 5 | 5 | 5 | 5 |
| ancho | 4 | 4 | 4 | 4 | 4 |
| ialu | 5 | 5 | 5 | 5 | 5 |
| fpalu | 4 | 4 | 4 | 4 | 4 |
| caché L1 | 128 | 256 | 128 | 128 | 256 |
| caché L2 | 2048 | 2048 | 2048 | 2048 | 2048 |
| caché L3 | 2048 | 2048 | 2048 | 2048 | 2048 |

Aquí podemos ver todos los valores que hemos usado para ejecutar los benchmarks y los respectivos cambios que hemos decidido implementar. Primero hemos hecho las pruebas con los cambios de manera individual y por último hemos hecho una quinta prueba aplicando los 3 cambios decididos anteriormente. Los cambios aplicados han sido aumentar la caché L1, la ventana de instrucciones y la cantidad de instrucciones que entran y de decodifica. Más adelante a medida que se vayan viendo los cambios se irán explicando y por qué su decisión la hipótesis respecto a dichos cambios y los resultados y unas pequeñas conclusiones.

Con los valores predeterminados hemos obtenido los siguientes resultados en los benchmarks.



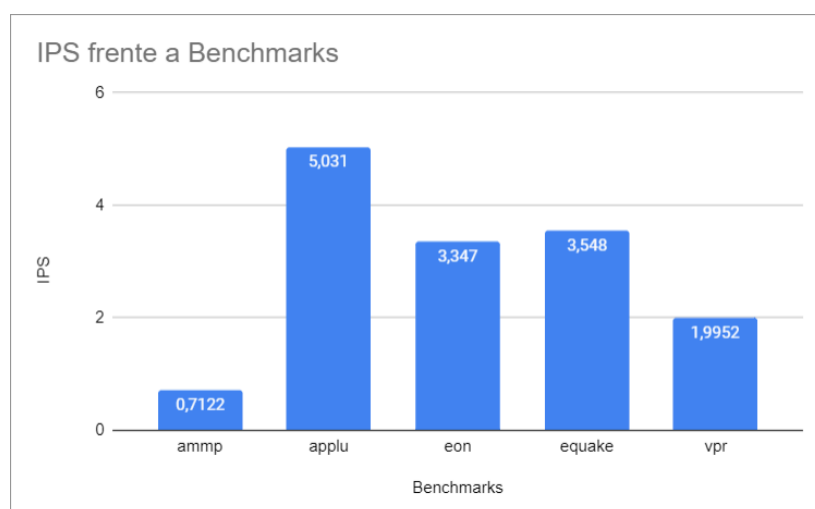
Como podemos ver en los benchmarks donde hemos sacado más rendimiento a las instrucciones por segundo han sido el applu, eon y equake. Siendo el applu donde sacamos más rendimiento.

1R CAMBIO

En el primer cambio he decidido aumentar la caché L1 de 128 a 512.

La razón de este cambio es ver si duplicando la capacidad de la caché más rápida nos permite almacenar más información a costa de que tardará más en buscar la información que necesite cuando la vaya a utilizar.

Cache L1 Duplicada.

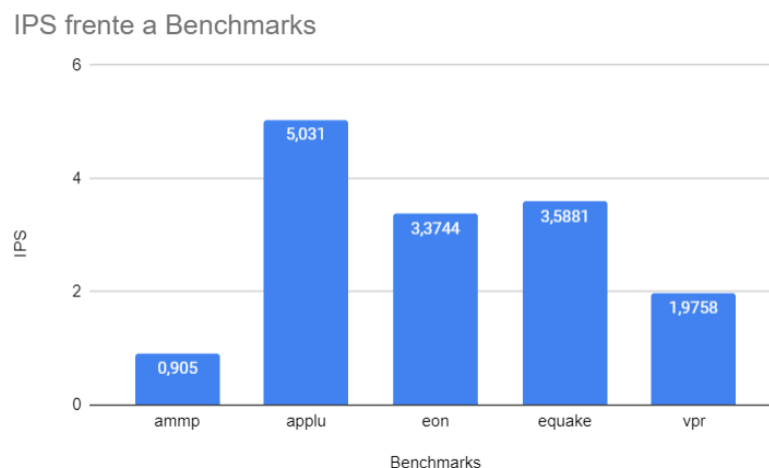


Como resultado no hemos obtenido ningún cambio por lo que se nos dificulta saber si nuestra hipótesis de sacrificar algo de velocidad en buscar información a cambio de aumentar la capacidad de almacenaje nos beneficia o no, por lo que me tomaré la

libertad de expandir más la hipótesis. A partir de no obtener ningún cambio tanto positivo como negativo creo que en caso de querer mejorar la capacidad de la caché se debería aumentar la L3 o como mucho la L2 ya que la L1 lo que priorizamos es completamente la velocidad en vez de la capacidad.

2do CAMBIO

En este caso he decidido aumentar la ventana de instrucciones, que es el conjunto de instrucciones que el procesador puede analizar y ejecutar de manera simultánea. Por lo que creo como hipótesis que podría aumentar el paralelismo y con eso su rendimiento a la hora de ejecutar instrucciones ignorando que requeriría un coste energético bastante superior.

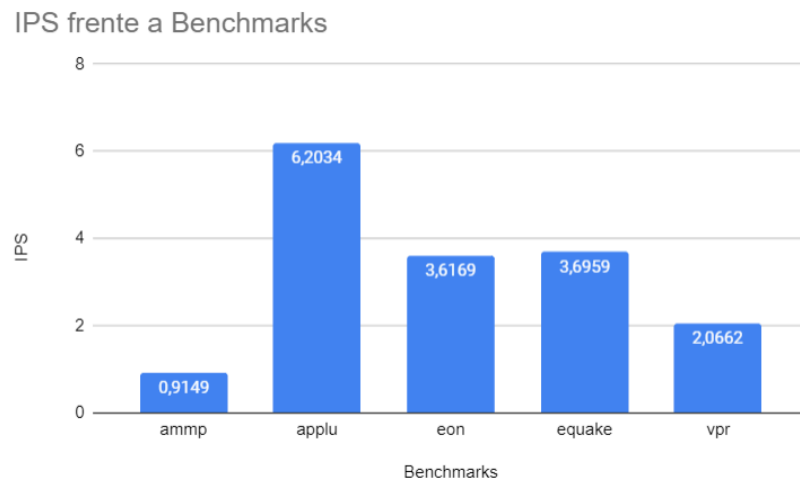


Una vez obtenemos los resultados podemos ver que únicamente hemos obtenido una mejora ligeramente notable en el benchmark ammp, y una mejora aún más sutil en el eon y el equake. Por lo si que mejora ligeramente su rendimiento pero no lo suficiente como para llevar esa mejora a cabo ya que sería un cambio muy costoso como para lo que proporciona de mejora.

3r CAMBIO

En este caso he decidido ir a por algo más seguro que es el aumentar el número de instrucciones que pueden entrar y se pueden decodificar es decir, aumentar el fetch y el decode de 6 a 12. En mi hipótesis permitirá recibir y gestionar muchas más instrucciones por lo que tendremos una mejora de rendimiento bastante notable ya

que nos permitirá una gestión de las instrucciones y un paralelismo superior a la anterior.

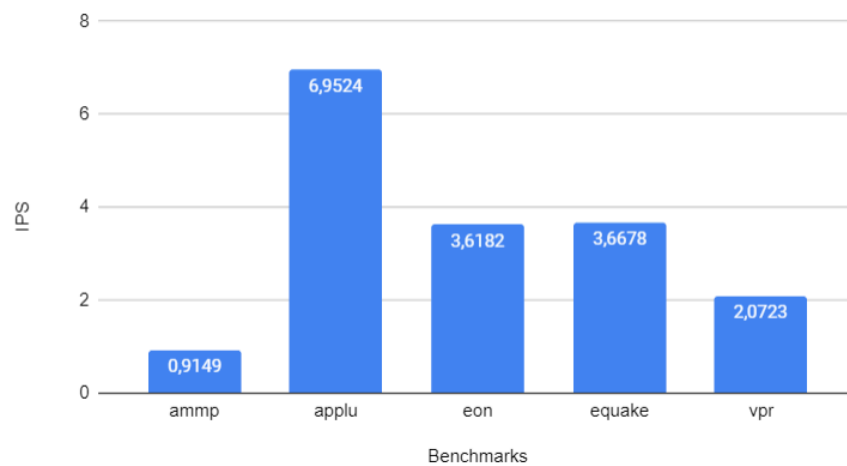


Y como podemos observar hemos sacado un rendimiento mayor y notable en todos los benchmarks por lo que realmente sí que beneficia centrar las mejoras en paso de más instrucciones y su gestión y paralelismo. Y podemos corroborar que nuestra hipótesis era correcta.

Aplicamos los 3 cambios anteriores a la vez

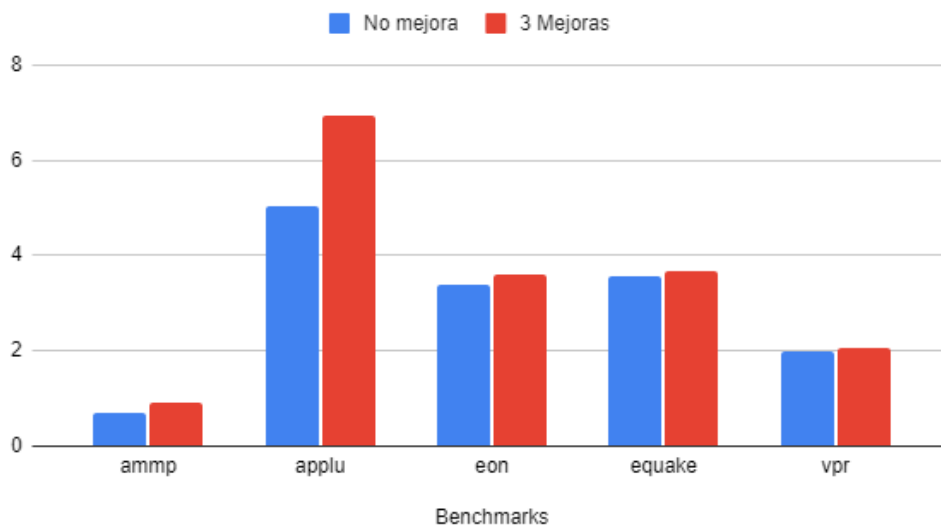
Ahora en este caso con todos los cambios a la vez creo que nos proporcionarán una mejora bastante notable ya que ya de por sí cada mejora era notoria menos la de la caché en este caso si será de ayuda sobre todo hará sinergia con la mejora de la del fetch y decode ya que permitirá que todas las instrucciones de más que están llegando puedan ser almacenadas con mayor facilidad y la mejora de la ruu nos permitirá gestionar más cantidad y nos proporcionará mayor paralelismo con todas las instrucciones de más que nos llegan.

IPS frente a Benchmarks



Y como podemos observar todo lo deducido en la hipótesis ha sido correcto ya que la sinergia entre las 3 mejoras han sido más que satisfactorias ya que hemos obtenido mejoras en todos los benchmarks sobre todo en el applu.

No mejora y 3 Mejoras



AMD Ryzen 5 7600X

Per poder millorar el rendiment del processador AMD escollit hem procedit a modificar els paràmetres següents:

- Hem pogut observar que el miss rate que tenia la caché de segon nivell era molt elevat així que hem optat per duplicar tots els seus paràmetres. En augmentar aquestes característiques passarem a tenir més espai, reduïrem aquest % de falla comportant així un augment del IPC.
- Hem augmentant la associativitat de les caches de primer nivell. L'associativitat d'una memòria caché és la manera en la que assigna les entrades de la caché a les direccions de memòria, és a dir, és el que determina quantes línies en la caché poden emmagatzemar dades amb el mateix index. Quan la associativitat millora/augmenta, dona pas a una major flexibilitat en la assignació de direccions de memòria, cosa que redueix els conflictes i per tant millora el percentatge de hits.

No s'ha pogut millorar mitjançant un tercer paràmetre ja que tots els que s'han intentat canviar no han donat uns resultats millors als ja obtinguts. Havíem augmentat el tamany de la finestra d'instruccions per poder executar més instruccions paral·lelament però no ha donat fruit a l'igual que tampoc ni augmentar ni disminuir el tamany de la cua d'accés a memòria. Per la cua d'accés a memòria l'objectiu era tenir un % d'ocupació més elevat que l'original (lsq_full) ja que aquest era null en molts dels benchmarks i estavem perdent. També s'ha provat l'execució proporcionant-li més ALU's per reduir la quantitat d'instruccions que s'esperen esperant a poder entrar a les unitats funcionals per a ser executades. El problema de posar tantes ALUs és que podem augmentar el consum d'energia del processador limitant-nos el rendiment.

Ja per últim s'ha intentat considerar una amplada de banda més gran per poder transferir més dades i una latència menor ja que aquest és el temps que tardem a accedir a elles.

L'IPC no ha variat gaire, i en el cas del Ammp ha empitjorat bastant.

Ens han quedat les dades de la següent manera:

| | |
|-----------------------------|---|
| Dades Caché L1 Instruccions | Tamany total 32 KB, 16 assoc, 32 tamany bloc, 64 sets |
| Dades Caché L1 Dades | Tamany total 32 KB, 16 assoc, 32 tamany bloc, 64 sets |
| Dades Caché L2 | Tamany total 2 MB, 16 assoc, 64 tamany bloc, 2048 sets |

Resultats de les simulacions:

| <i>Benchmark</i> | <i>ammp</i> | <i>applu</i> | <i>eon</i> | <i>equake</i> | <i>vpr</i> |
|------------------|-------------|--------------|------------|---------------|------------|
| <i>IPC</i> | 1,0088 | 3,4388 | 2,6836 | 2,8834 | 1,8209 |

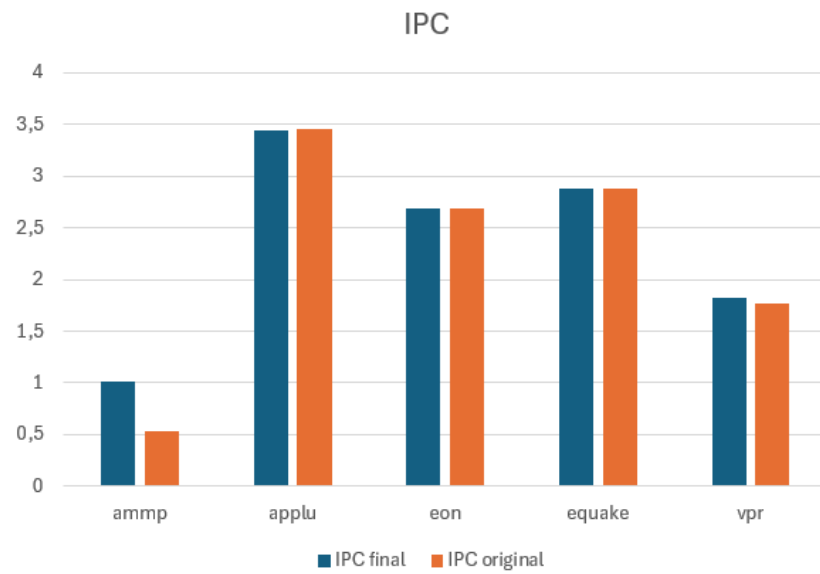
| | <i>ammp</i> | <i>applu</i> | <i>eon</i> | <i>equake</i> | <i>vpr</i> |
|----------------------|-------------|--------------|------------|---------------|------------|
| <i>miss_rate_d11</i> | 0,1236 | 0,0068 | 0,0018 | 0,0004 | 0,0126 |
| <i>miss_rate_il1</i> | 0,000 | 0,000 | 0,000 | 0,000 | 0,000 |
| <i>miss_rate_ul2</i> | 0,0090 | 0,2542 | 0,0115 | 0,2897 | 0,1231 |

Fracció de cicles en els que la cua d'accés a memòria està plena amb la millora:

| <i>Benchmark</i> | <i>ammp</i> | <i>applu</i> | <i>eon</i> | <i>equake</i> | <i>vpr</i> |
|------------------|-------------|--------------|------------|---------------|------------|
| <i>lsq_full</i> | 0,6949 | 0,0357 | 0,083 | 0,0000 | 0,001 |

Fracció de cicles en els que la cua d'accés a memòria està plena sense la millora:

| <i>Benchmark</i> | <i>ammp</i> | <i>applu</i> | <i>eon</i> | <i>equake</i> | <i>vpr</i> |
|------------------|-------------|--------------|------------|---------------|------------|
| <i>lsq_full</i> | 0,8325 | 0,0000 | 0,0000 | 0,0000 | 0,001 |



Conclusió

En les dues millores el canvi vist ha estat bastant petit, podríem arribar a la conclusió que els processadors d'avui en dia son molt potents i tenen un rendiment molt elevat deixant així el marge de millora bastant petit. Aconseguir quadrar la informació de totes les pàgines webs disponibles ha estat tot un repte ja que no tots mostraven totes les dades iguals.

Comentari del Paràgraf

"For now, AMD continues to pressure Intel in the value gaming segment: AMD's new Ryzen 7 5700X3D provides up to 30% more gaming performance than the lackluster Core i5-14400, making it a compelling option for those obsessed with the fastest value CPU for gaming, but it comes with tradeoffs in productivity apps. In contrast, AMD's Ryzen 5 7600X and Ryzen 5 7600 are better all-rounders and sell for much less than before. Pricing for AMD's motherboards and DDR5 memory is also now more favorable, giving the Ryzen lineup a win over the Core i5-14400."

Aquest paràgraf parla sobre la continua competició en la que es troben Intel i AMD pel sector dels videojocs a través dels seus processadors. Comenta com AMD proporciona un rendiment de joc bastant superior a l'altre marca per poder atraure als jugadors més professionals encara que això comporta la seva part negativa. Ve amb limitacions en les aplicacions de productivitat.

Destaca com dos dels seus processadors Ryzen, les plaques base i la memòria DDR5 es venen per un preu inferior al que tenien abans donant a la línia Ryzen d'AMD cert avantatge sobre el Core i5-14400.

Un augment del 30% en el rendiment de joc és significatiu i pot ser un factor determinant per molts compradors. Els usuaris han de parar i reflexionar sobre les seves prioritats i necessitats, saber si els surt a compte sacrificar rendiment d'aplicacions per jugar per els preus que ofereixen o no.

WEBGRAFIA

<https://ark.intel.com/content/www/us/en/ark/products/236788/intel-core-i5-processor-14400-20m-cache-up-to-4-70-ghz.html> (Pagina principal de intel)

<https://www.techpowerup.com/cpu-specs/core-i5-14400.c3449> (techpowerup)

<https://www.quora.com/How-do-I-find-out-the-IPC-instructions-per-cycle-of-a-CPU-prior-to-buying-the-CPU-For-example-is-there-any-website-that-gives-a-direct-IPC-as-compared-to-the-score> (Info de como encontrar el IPC de una cpu, 4 por core)

https://docs.google.com/spreadsheets/d/18ln8SKlGRK5_6NymgdB9oLbTJCFwx0iFI-vUs6WFyuE/edit?gid=1076260513#gid=1076260513 Especs de procesadores)

<https://drive.google.com/drive/u/0/folders/1W4CIRKtNML74BKjSbXerRslzAUk3ppSG> (Micro arquitecturas)

[Documentation for AMD Processors, Accelerators, and Graphics](#) (Documentació oficial AMD)

[AMD's Zen 4 Part 1: Frontend and Execution Engine – Chips and Cheese](#)

[ualberta.ca](#) (Amplada de banda i latència)

[Hot Chips 2023: Characterizing Gaming Workloads on Zen 4 – Chips and Cheese](#)

[AMD Ryzen 5 7600X Specs | TechPowerUp CPU Database](#) (Informació del processador)