Universidad de Costa Rica
Escuela de Ciencias de la Computación e Informática

Fundamentos de Arquitectura

Examen Parcial #1

Dylan Gabriel Tenorio Rojas C07802

Profesor
Francisco Arroyo

8 Octubre de 2021

De acuerdo con Napoles, introduce la memoria caché de la siguiente forma: La memoria caché es una memoria pequeña, rápida y especial de alta velocidad, se ubica entre la CPU y la memoria principal, diseñada para acelerar el procesamiento de instrucciones del microprocesador, el cual, puede acceder a los datos almacenados en la caché mucho más rápido que a aquellos datos almacenados en la memoria RAM. (2013, p. 14). Así como lo indica Nápoles, esta caché es una parte fundamental de las computadoras por su función de alojar datos que serán o fueron procesados por la CPU (Central Process Unit). Además, dicha memoria también aloja las instrucciones de futuros procesos que llevará a cabo la CPU, de forma que no sólo acelera la transmisión de datos entre la memoria y el procesador, sino que provee alojamiento a instrucciones que permitirán realizar cálculos y enviar señales hacia las demás partes de la computadora. La razón por la cual esta memoria es tan ágil a la hora de manejar datos es debido a su cercanía al procesador, pudiendo incluso estar contenido en el CPU, lo cual permite que la comunicación entre el procesador y la memoria caché sea más veloz que la comunicación con la memoria principal, la cual es la RAM; esto le permite a la memoria caché almacenar datos de uso frecuente o de gran valor y que a su vez esté disponible de primera mano al procesador, incrementando tanto el rendimiento como la potencia del procesador.

Arquitectura de uso

1. Memoria Caché L1

La memoria caché L1 es el tipo de memoria caché más cercana al procesador y a su vez, la memoria más pequeña de todas. Esta memoria se encuentra principalmente directamente en el procesador y en caso de ser un procesador multinúcleo, esta se encuentra fuera del procesador. De acuerdo a Sathyanarayanan (s. f.), por la jerarquía establecida en estos niveles de memoria caché, la memoria L1 consulta primaria y fundamental del procesador. Esta también será la de menor tamaño (entre 2 KB y 32 KB) y la más veloz, puesto que por encontrarse tan cerca del procesador, funcionará a la velocidad del reloj, la más veloz de toda la computadora. Esta memoria caché posee dos funciones principales: almacenar instrucciones del futuras ejecuciones de programa y almacenar datos de uso frecuente por el procesador. Estas funciones no las realiza en conjunto, sino que estas son llevadas a cabo por dos subtipos de esta memoria, con el fin de optimizar los procesos de escogencia de la información por parte del CPU. Estos dos subtipos se conocen como L1 Data Cache y L1 Instruction Cache.

- L1 Data Cache

Esta memoria caché se encarga del almacenamiento de datos provenientes de procesos realizados por el procesador, Este subtipo trabaja junto a la L1 Instruction Cache en los procesamientos de información que le llegan al procesador. Por ejemplo, una instrucción almacenada en la L1 Instruction Cache podrá incluir tareas

que requieran mover o alojar datos en la L1 Data Cache. Este subtipo funciona de manera similar a los registros internos del CPU, con la principal diferencia de que esta memoria caché es más grande que el registro y solamente trabaja con los datos próximos a procesar, con datos obtenidos de ejecuciones de programa o de datos utilizados en cálculos de un proceso.

- L1 Instruction Cache:

La memoria L1 Instruction Cache se encarga de guardar las instrucciones de las ejecuciones de un programa. De manera similar a los registros, esta memoria caché aloja instrucciones en lenguaje máquina que se utilizará en el procesamiento de información o manejo de los componentes de una computadora. Posee de la misma forma, una capacidad superior a la de los registros internos del CPU y es utilizada principalmente para instrucciones próximas o instrucciones des largos que no pueden ser almacenadas únicamente en los registros internos. A su vez. trabaja de la mano con los registros y la L1 Data Cache, agilizando procesos y teniendo una interconexión semejante a la velocidad del mismo procesador.

La comunicación entre la memoria caché L1 y el procesador fue abarcada dentro de cada subtipo de la caché L1 y esta generalidad permite el procesamiento veloz de información. No obstante, la memoria caché no posee tanta capacidad como para poder almacenar todo tipo de información y brindarla con la misma velocidad a la que ya transmite datos. Este problema no permite expandir la L1, pues esto afectaría negativamente el tiempo de respuesta de la transferencia de datos. Por esto y como toda jerarquía, basa su velocidad solo para datos importantes y en caso de requerir mayor espacio para alojar datos, la memoria caché L1 solicita espacio directamente al siguiente nivel de la jerarquía, la memoria caché L2, con la que tiene una conexión bilateral en donde se movilizan los datos dependiendo del procesamiento actual.

2. Memoria Caché L2

Acorde a lo mencionado por Napoles (2013, p. 48-49), la memoria caché L2 es el siguiente nivel en la jerarquía de las memorias caché y presenta las características principales de que su capacidad de almacenaje aumenta a costo de la velocidad de transferencia de datos y que, a diferencia del primer nivel, la memoria caché L2 no se encuentra directamente en el núcleo del procesador y es conectada al CPU mediante un bus de alta velocidad. Este nivel se encuentra inmediatamente antes del acceso a la memoria principal y por esta razón, se le llama SRAM (Static Random Access Memory), aunque no pertenece a la RAM y es de menor tamaño que está.

La comunicación que posee la memoria de primer nivel con la caché L2 es de forma jerárquica, por lo que siempre el procesador habrá consultado sus registros y la caché L1 con antelación para poder acceder a la caché L2. De esta forma, se asegura obtener el mejor rendimiento de procesamiento. A pesar de esto, la

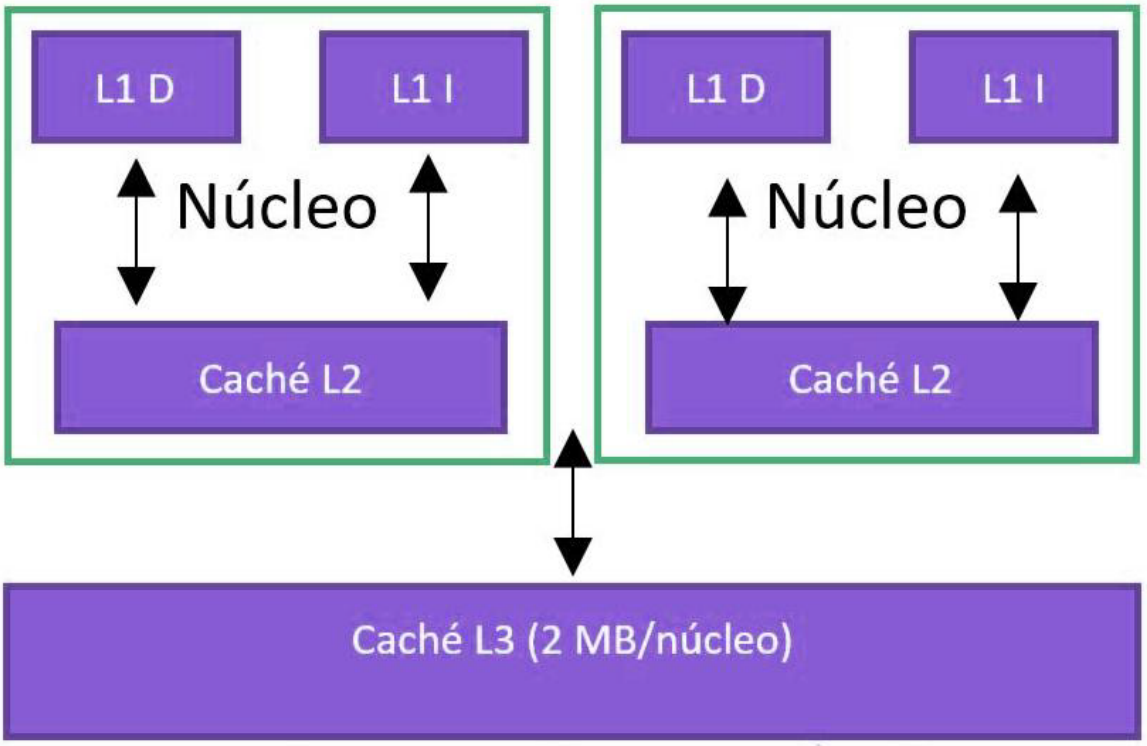
velocidad del procesamiento sigue estando caché L1 y de estas es que depende directamente la transferencia de la caché L2. (Napoles, 2016, p. 48-49).

Esta caché es fundamental en la transferencia de datos, principalmente con la memoria principal. Su potencia le permite cargar datos de forma mucho más rápida que utilizar directamente el espacio en memoria de la RAM para la realización de los procesos enviados al procesador. Muchos nuevos computadores incluyen este tipo de caché directamente conectada al procesador, haciendo más veloz este nivel. Como características principales, se pueden nombrar su capacidad de entre 256 KB y los 4 MB, una cantidad considerablemente más grande de espacio que la caché L1; su proximidad a la RAM y que estas trabajan en conjunto para transmitir datos de la manera más óptima posible y finalmente, su alta velocidad de transmisión, aunque sigue por debajo de la memoria caché L1.

3. Memoria Caché L3

La memoria caché L3 forma parte de la jerarquía de memorias caché que posee un computador. Así como los anteriores niveles, la caché L3 almacena datos de procesos ejecutados, ejecutándose o por ejecutar y la principal diferencia entre esta caché y los niveles anteriores es que esta es compartida por todos los núcleos del procesador. Sathyanarayanan (s.f.) nos indica que entre sus principales características, se encuentra su capacidad, superior a la caché MB y 8 MB de almacenamiento; su ubicación es de forma paralela a la RAM, por lo que trabajan en conjunto para brindar información no encontrada en los niveles L1 y L2, y que su principal función consiste en ser un soporte final de información antes de proceder a los almacenamientos más lentos de toda la computadora. El orden en el que se encuentra, sigue un camino similar al que ocurre en la caché L2: el dato primeramente es consultado por el registro interno del procesador y si no se encuentra en este, avanza hasta las caché L1 en cada núcleo; una vez consultado en el primer nivel de las cachés y el dato no fue hallado, se procede a consultar al nivel 2 de las cachés de cada uno de los núcleos y si finalmente el dato no se encuentra aquí, el dato pasa a ser consultado directamente a la memoria RAM de forma paralela a la caché L3.

Esta caché cumple un papel vital en el procesamiento y transferencia de datos, porque desprende su capacidad de alojamiento a todos los núcleos del procesador para que logren efectuar sus procesos de la manera más óptima posible, sin entrar directamente a las memorias más lentas. Probablemente, la memoria caché con mayor complejidad que existe y a su vez, una de las más importantes en brindar los datos a una velocidad donde la comunicación con el procesador no se quede rezagada por las diferencias en las velocidades de estas.

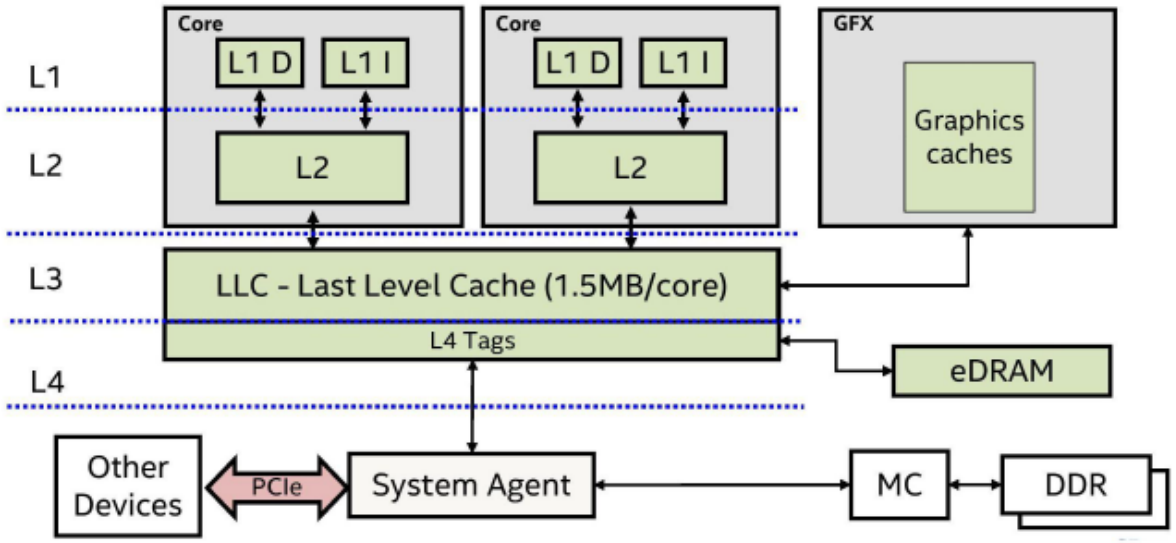


Con este ejemplo podemos entender como funcionan las anteriores 3 memorias con respecto a su conexión al CPU o sus núcleos.

4. Memoria Caché L4

La memoria caché L4 es un tipo nivel de caché que utiliza el procesador para almacenar datos, igual a los anteriores niveles. Este nivel es bastante reciente y se enfoca principalmente en el manejo de datos entre la memoria secundaria y muchas veces también en la tarjeta gráfica, que similar al procesador realiza cálculos variados y requiere de celdas en memoria para alojar distintos datos mientras realiza la ejecución.

Este nivel podría considerarse una variación de la memoria caché L3, puesto que su función es la de apoyo directo a los registros y al procesador en conjunto, por lo que es utilizada como último escalón antes de la consulta a la memoria secundaria e incluso a memorias terciarias como serían las memorias extraíbles. Por esto y de acuerdo a su función de apoyo, muchas veces se cataloga a esta memoria como un subconjunto de la memoria caché L3 y es posible que la caché L3 cumpla funciones que requerirían una caché L4 pero en su ausencia, esta asuma ese rol. Cuando esta memoria caché se encuentra en el sistema de la computadora. Con el siguiente ejemplo podemos ver como funciona la L3 junto con las demás memorias.



Elementos lógicos

- Direcciones de memoria

Las direcciones de memoria podemos encontrarlas en 2 formas, virtual o física, la memoria virtual nos sirve para que los programas pueden dirigirse a memoria desde la lógica sin preocuparnos de la cantidad de memoria física disponible.

- Tamaño de la caché

El tamaño de la caché es un aspecto fundamental en el desempeño que pueda ejercer en el almacenamiento de datos y transferencia de estos al procesador. Lo fundamental de encontrar el tamaño más óptimo de la caché se define según sea la memoria principal y las características del procesador al que va a apoyar. Sin embargo, su capacidad no puede sobrepasar cierto límite pues a mayor capacidad, también se pierde potencial de velocidad y esto afecta negativamente el desempeño del procesador en las tareas realizadas.

- Función de correspondencia

Es la forma en la que la memoria principal puede transmitir datos hacia la memoria caché, pues esta última no tiene la capacidad de asignar la información 1 a 1, debido a su tamaño. Esta función puede ser de tres tipos:

1. Correspondencia Directa:

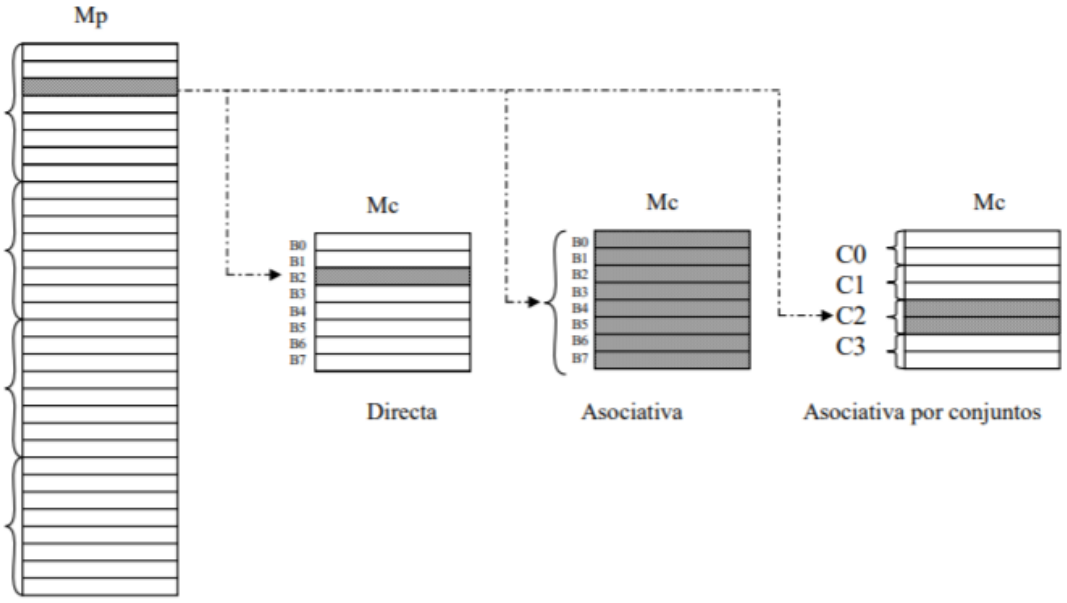
donde se asignan bloques de memoria principal a una línea de memoria caché, con el problema de que toda la memoria principal estaría etiquetada en la memoria caché.

2. Correspondencia Asociativa:

Que permite que cualquier bloque pueda ser etiquetado en la caché sin crear ambigüedades pero su condiciones se vuelven muy complejas.

3. Correspondencia Asociativa por conjuntos:

Que se trata de un híbrido entre las anteriores funciones de correspondencia. En ella se divide la caché en diferentes conjuntos, que constan cada uno de un número determinado de bloques de caché (un bloque de caché = una vía). Se seleccionará uno de los conjuntos a través de una función de correspondencia directa y una vez que se haya hecho, se elegirá uno de los bloques incluidos en él a través de una función de correspondencia totalmente asociativa.



- Algoritmo de reemplazamiento.

Su función es la de seleccionar un bloque de caché de entre un grupo de ellos, de acuerdo a un determinado criterio que va a depender del método de reemplazamiento utilizado. Este algoritmo es muy importante y habrá que usarlo cada vez que haya que sustituir un bloque de caché por otro. La función de reemplazamiento sólo tiene sentido cuando se utilizan funciones de correspondencia asociativa por conjuntos o totalmente asociativa, ya que si estamos utilizando correspondencia directa, solamente existiría un único bloque candidato para ser sustituido. A medida que va aumentando el tamaño de la caché, la función de reemplazamiento tiene una menor importancia, porque la tasa de aciertos va a ser muy alta. Existen una gran diversidad de funciones de reemplazamiento. Las más utilizadas son: Aleatoria, LRU, FIFO, LFU. De entre todas ellas, las más utilizadas son la aleatoria y la LRU, siendo ésta última bastante mejor que la aleatoria, cuando los tamaños de caché son pequeños.

1. Algoritmo de reemplazamiento aleatorio.

Con esta estrategia, elegiremos uno de los posibles bloques de forma aleatoria. Así conseguiremos que todos los bloques de caché sean sustituidos uniformemente. Este método resulta fácil de implementar, pero tiene el inconveniente de que bloques que han sido utilizados recientemente, o que son utilizados con mucha frecuencia, pueden ser sustituidos.

2. Algoritmo de reemplazamiento LRU.

Este método explota el principio de localidad temporal, que dice que bloques que han sido accedidos hace poco es probable que vuelvan a ser utilizados pronto. De entre los posibles bloques a sustituir se elegirá aquel que lleve más tiempo sin ser utilizado. Esta estrategia requiere de un hardware más complejo, ya que ha de registrarse la última vez que fue accedido un bloque.Cuando el número de bloques de caché es muy grande, esta estrategia se encarece bastante, ya que hay que hacer un gran número de comparaciones al hacer la selección.

3. Algoritmo de reemplazamiento FIFO (1º en entrar 1º en salir).

En este método el bloque sustituido será aquel que llegó antes a la caché.

4. Algoritmo de reemplazamiento LFU.

En este método el bloque a sustituir será aquel al que se acceda con menos frecuencia. Habrá que registrar por lo tanto, la frecuencia de uso de los diferentes bloques de caché.

- Políticas de escritura

Esta característica se define como la forma en la que la caché y la memoria principal interactúan con la información dada. De esta, aparecen dos tipos donde en uno se realiza la escritura en caché de manera simultánea en la RAM o que puedan ser cargadas en otro momento de ejecución y se mantienen actualizadas únicamente en caché.

- Tamaño de línea en caché

Así como el tamaño de la caché, es muy importante el tamaño de cada línea de almacenaje que posee la caché. Brindar un tamaño estándar es muy complicado debido a que en los inicios de uso de la caché, los datos serán los más frecuentes y será más sencillo y rápido la consulta de los datos en memoria pero conforme se van incluyendo más líneas, el proceso se hace más complejo y muchas direcciones no serán utilizadas, por lo que se pierde potencial en la transmisión de datos. Así como con el tamaño de la caché, entre más bloques de memoria existan, más datos pueden ser almacenados pero se arriesga velocidad y exactitud en la solicitud de datos, y viceversa.

- Hit rate

La tasa en la que la memoria caché almacena correctamente el dato en la dirección correspondiente.

Elementos Estructurales

Block: La unidad mínima que se puede transferir entre la memoria principal y la caché.

Frame: Es un término que se usa para diferenciar entre el dato transmitido y el espacio de memoria física.

Line: Una parte de la cache para guardar 1 bloque de memoria.

Tag: La forma en cómo nos podemos referir a la dirección de una líne.

Determinación de “miss” o “hit”

- Acierto de caché (hit): el contenido de la dirección se encuentra en un bloque ubicado en una línea de la caché.
- Fallo de caché (miss): el contenido de la dirección no se encuentra en ningún bloque ubicado en alguna línea de la caché.

Si en la ejecución de un programa se realizan Nr referencias a memoria, de las que Na son aciertos caché y Nf fallos caché, se definen los siguientes valores:

- Tasa de aciertos: Ta = Na/ Nr
- Tasa de fallos: Tf = Nf/ Nr
- Evidentemente se cumple: Ta = 1 - Tf

Operación de lectura

En una operación de lectura se lee la palabra completa de la RAM, es decir, la línea y la etiqueta. Si la etiqueta leída coincide con la procedente de la dirección física, significa que la línea contiene la palabra de Mp referenciada por dicha dirección física: se produce un acierto de caché. Si no coinciden las etiquetas, significa que Mc no contiene el bloque de Mp al que pertenece la palabra referenciada, por lo que se produce un fallo de caché.

Operación de escritura

Frente a aciertos en la caché existen dos alternativas: escritura directa y postescritura. Y frente a fallos en la caché otras dos: asignación en escritura y no asignación.

- Escritura directa o inmediata (write through)

Todas las operaciones de escritura se realizan en memoria caché y memoria principal

- Postescritura (copy back)

Las actualizaciones se hacen sólo en memoria caché
Se utiliza un bit de actualización asociado a cada marco de bloque para indicar la escritura del marco en memoria principal cuando es sustituido por la política de reemplazamiento

- Asignación en escritura (write allocate)

El bloque se ubica en memoria caché cuando ocurre el fallo de escritura y a continuación se opera como en un acierto de escritura, es decir, con write through o copy back

- No asignación en escritura (No write allocate)

El bloque se modifica en memoria principal sin cargarse en memoria caché

Medidas de Desempeño

Si frente a un fallo, el bloque de memoria principal se lleva a memoria caché al tiempo que la palabra referenciada del
bloque se lleva (en paralelo) a la CPU, el tiempo de acceso a memoria durante la ejecución de un programa será :
Tacceso = Na * Tc + Nf * Tp donde:
Na es el número de referencias con acierto
Nf es el número de referencias con fallo
Tc es el tiempo de acceso a una palabra de memoria caché
Tp es el tiempo de acceso a un bloque de memoria principal

El tiempo de acceso a un bloque de memoria principal, Tp, constituye la componente principal del tiempo
total de penalización por fallo.

El tiempo de acceso medio durante la ejecución del programa valdrá:
Tacceso_medio = Tacceso/ Nr = Ta*Tc + Tf*Tp donde:
Ta = Na/Nr es la tasa de aciertos
Tf = Na/Nr es la tasa de fallos
Nr es el número total de referencias a memoria
A la primera componente se le denomina Tacierto = Ta*Tc,

En cambio, si frente a un fallo, el bloque de Mp se lleva primero a Mc y después se lee la palabra referenciada de Mc y se lleva a la CPU.

El tiempo de acceso a memoria durante la ejecución de un programa será :
Tacceso = Nr * Tc + Nf * Tp y
Tacceso_medio = Tacceso/ Nr = Tc + Tf*Tp
En este caso Tacierto = Tc

REFERENCIAS

Cutress, I. (2020). A Broadwell Retrospective Review in 2020: Is eDRAM Still Worth It?. Recuperado de https://www.anandtech.com/show/16195/a-broadwell-retrospective-review-in-2020-is-edram-still-worth-it

Napoles, A. (2013). Análisis documental sobre memorias caché. Recuperado de http://ri.uaemex.mx/handle/20.500.11799/59160.

Smith, A. J. (2003). Cache memory. In Encyclopedia of Computer Science (pp. 180-187).

Smith, A. J. (1982). Cache memories. ACM Computing Surveys (CSUR), 14(3), 473-530.

Stallings, W. (2015). Computer Organization and Architecture Designing for Performance (10th ed., pp. 263-264). Prentice Hall.

Sathyanarayanan, A. Cache Memory Types Guide to Various Types Cache Memory. Recuperado