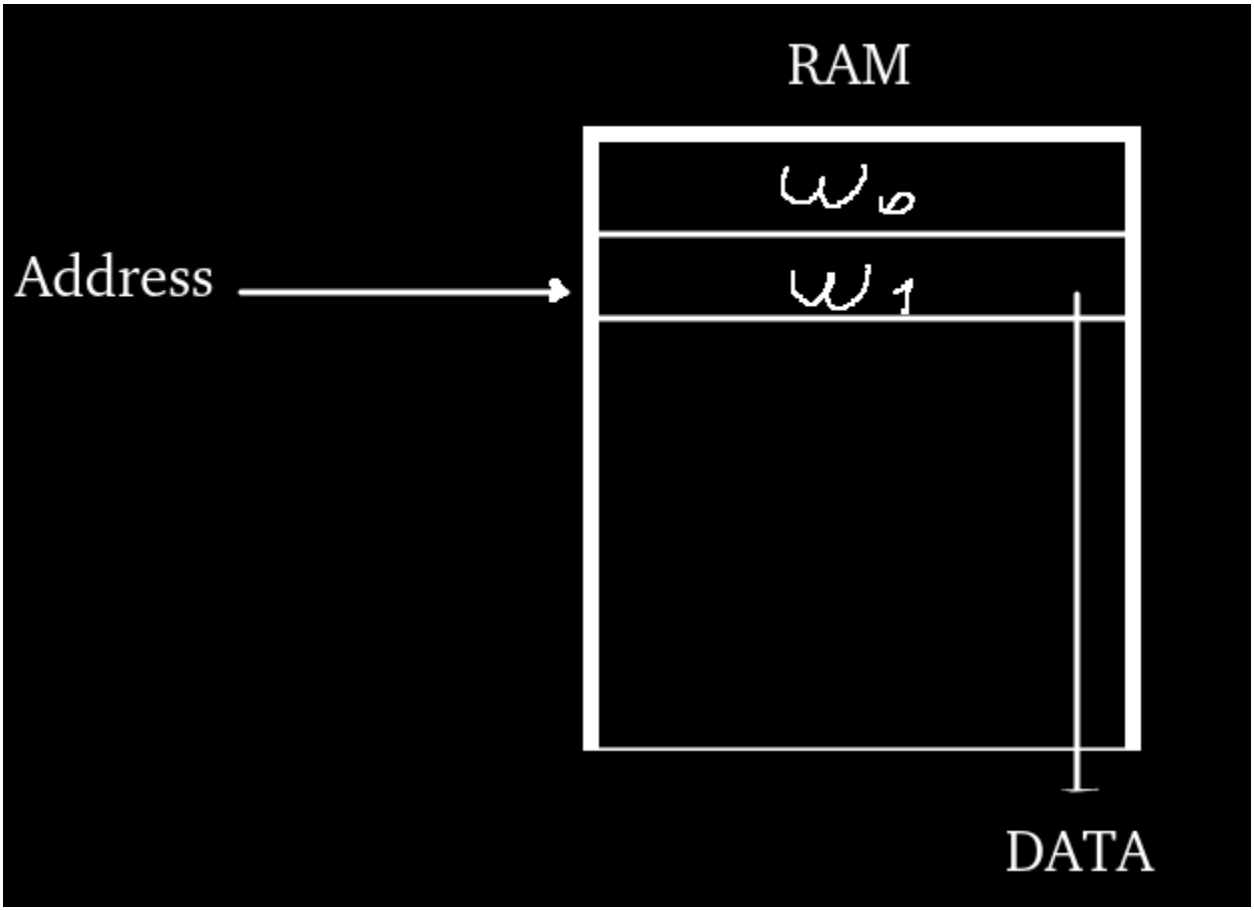
Memoria Cache

- ¿Como sabemos si la data esta presente?
- ¿Donde buscamos?

Memoria principal

RAM

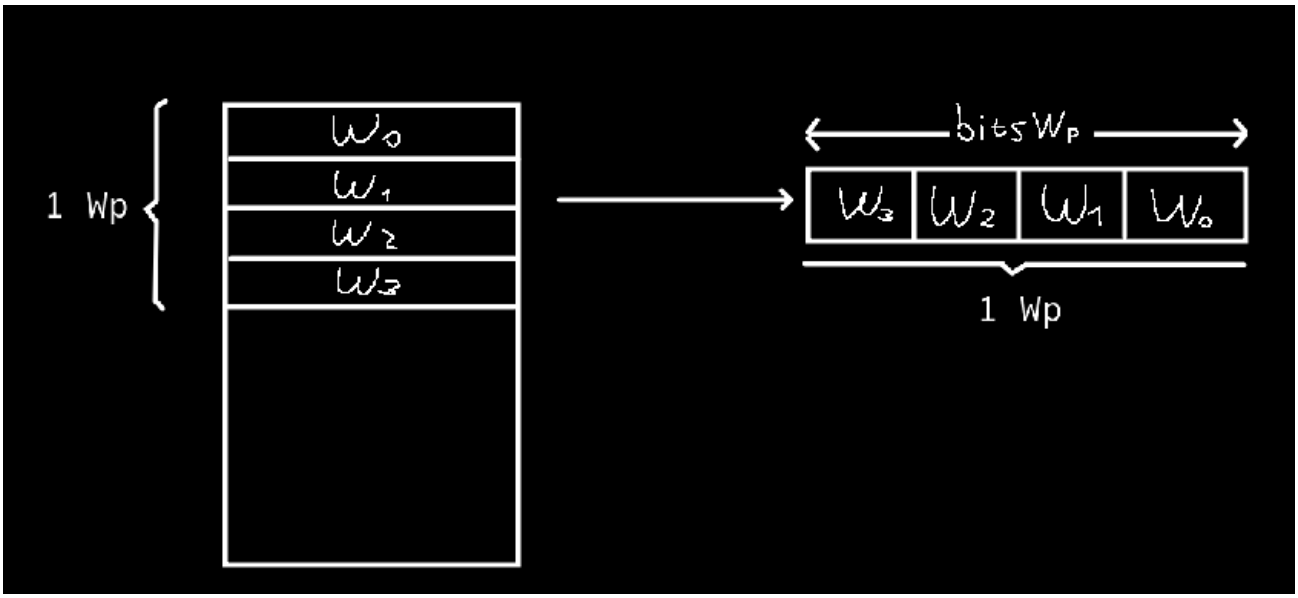


- Se accede a traves de una direccion
- Devuelve el contenido en esa direccion de memoria
- **Parametros de MemPrincipal**
  - **Cantidad de Wm**
  - **Tamaño de la palabra
  - **CapTotalMem = Wm * size(Wm)**

Palabras/Words de procesador y Mem Principal

- Las **Words de Memoria Principal** (Wm) tienen su tamaño (1byte si no se especifica)

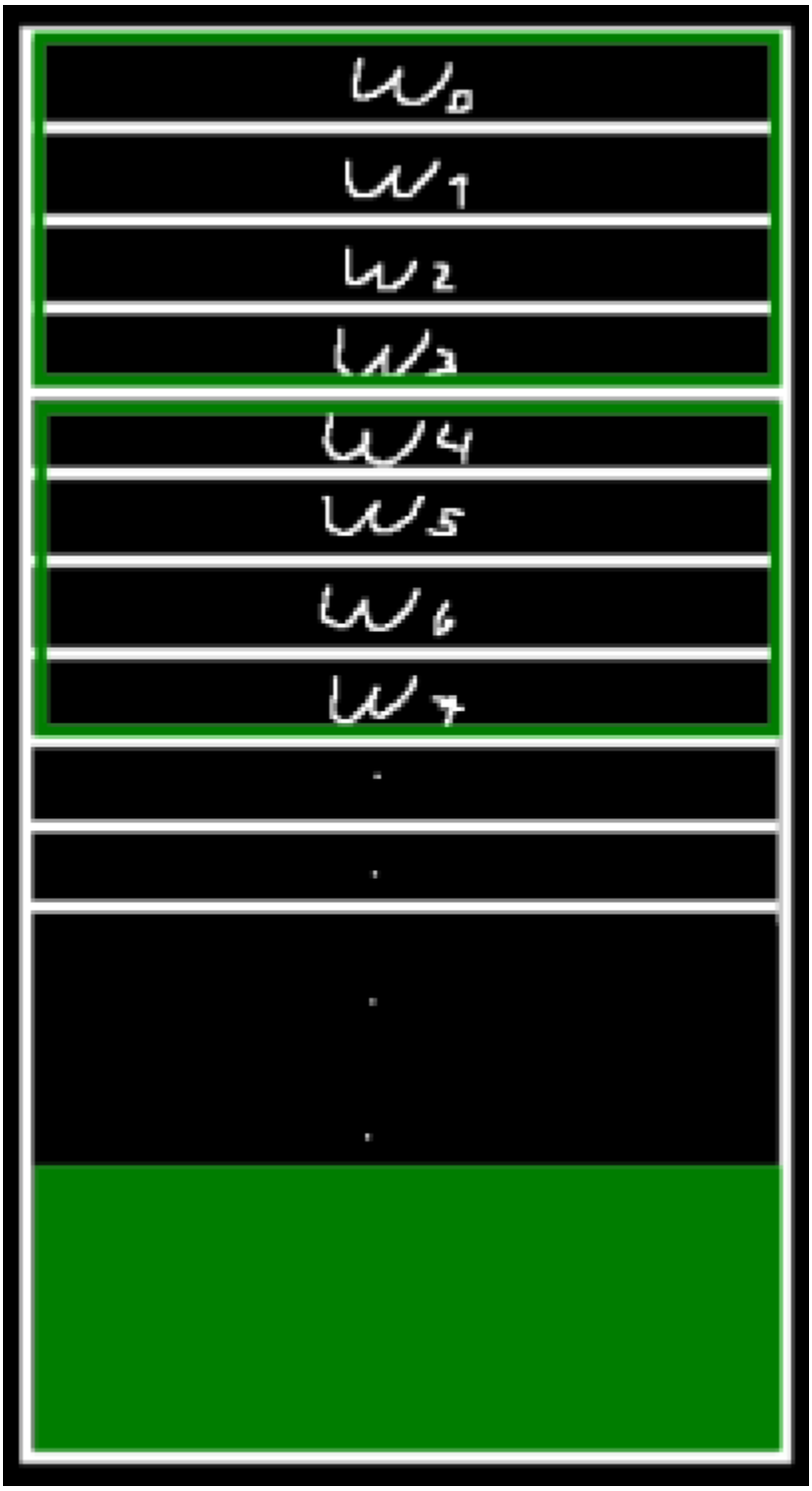
- Las **Words de procesador** (Wp) podrian tener igual o distinto tamaño a las de words de memoria (por lo general distinto)
- Por ejemplo si la Wp son de 32 bits
  - => 4 Wm seran una word del procesador
  - Por eso se le suma +4 al PC para acceder a la siguiente Wp

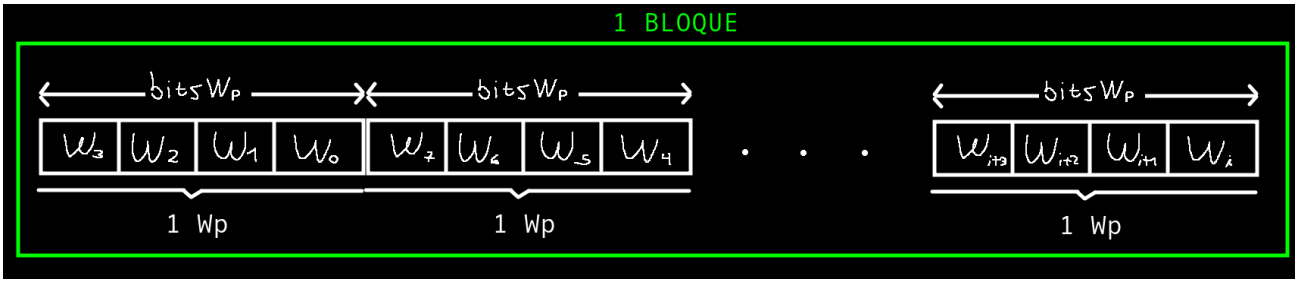


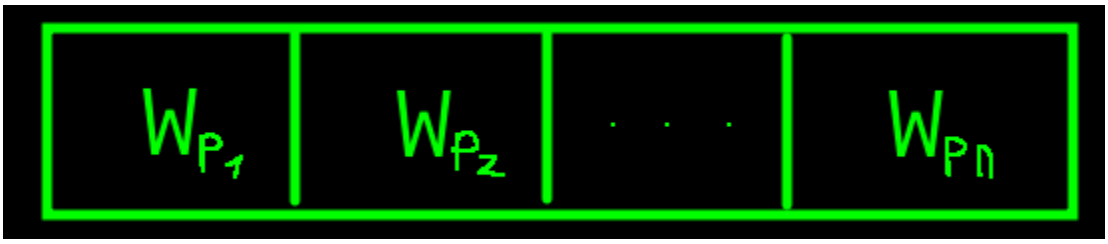
- ! Si la guia no aclara el tamaño de la Wp o Wm asumimos el LEGV8 !
  - ✅ Wp -> 64 bits
  - ✅ Wm -> 8 bits

Bloque

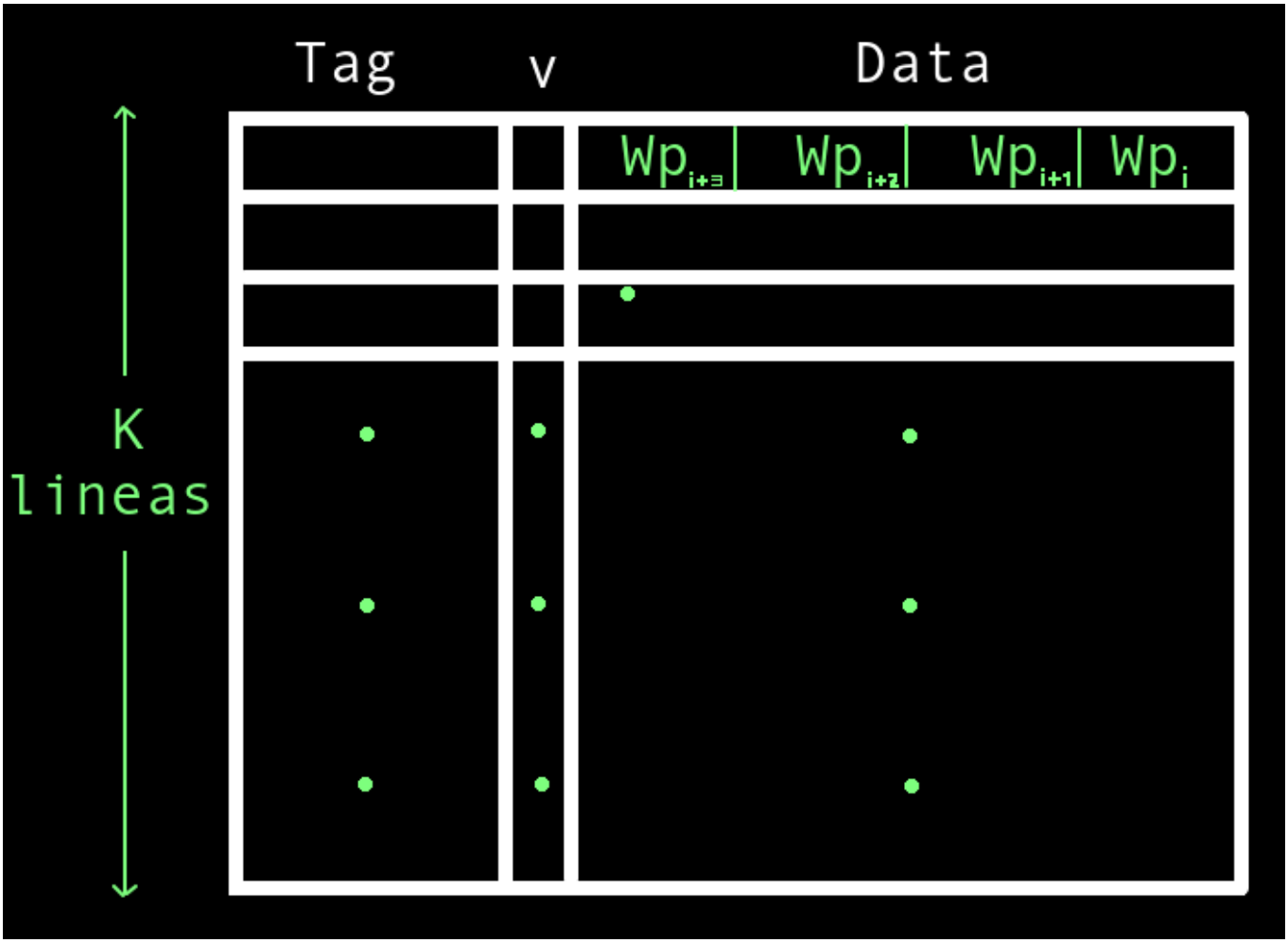
- Un conjunto de **Wp** es un solo bloque
- Entonces la Memoria principal la podemos ver organizada de a bloques
  - **Calculo cantidad de bloques
  1. Wp = Wm / size(Wp)
  2. Bloques = Wp / Cant_Wp_por_bloque


W0
W1
W2
W3
W4
W5
W6
W7


1 BLOQUE
bits Wp
bits Wp
bits Wp
W3 W2 W1 W0
W7 W6 W5 W4
Wi+3 Wi+2 Wi+1 Wi
1 Wp
1 Wp
1 Wp



Cache



- **Linea**
  - **Area de data** => Aloja un bloque (Conjunto de N Wp) (1 Bloque = 1 Linea)
  - **V (bit de validacion)** => 0 (No data) | 1 (data presente)
  - **Tag** => Identificador de la vez de memoria mapeada
  - **Calculo Cantidad de lineas **
    - Si tengo Capacidad total del area de datos puedo calcular la cant de L
    - CapTotalData = CantL * TamañoLineaData
- **Proceoso de busqueda** => Instantaneo (Concurrente)

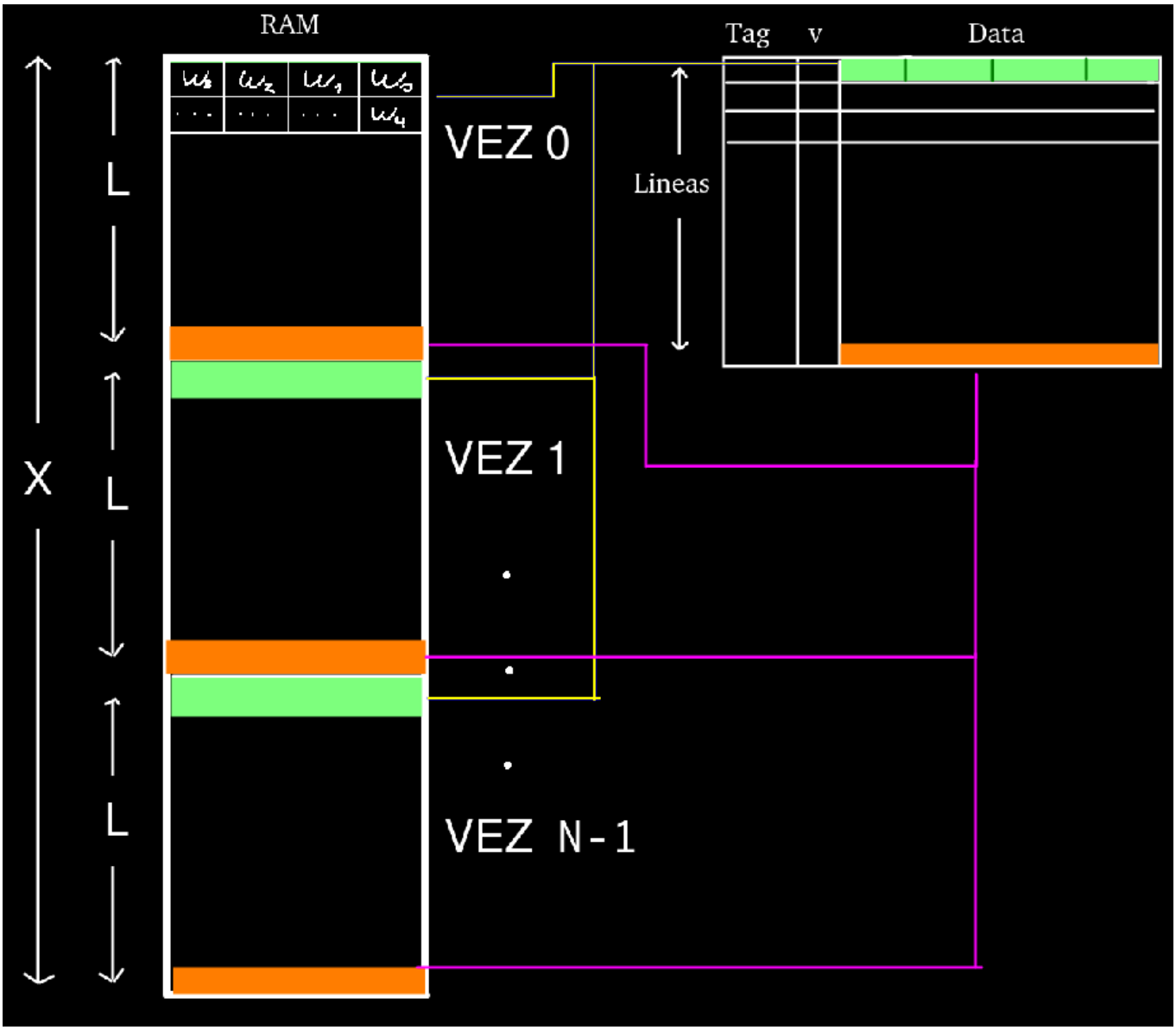
Distintios tipos de cache (Criterio de correspondencia)

- **X > L** => Siempre hay mas bloques de memoria principal que lineas de cache
- Debido a esto nace los criterios de correspondencia => Define el tipo de cache
- **¿ Como se corresponden X bloques de memoria principal en L lineas de cache?**
- Esto origina los distintos tipos de cache
- # Cache de mapeo directo

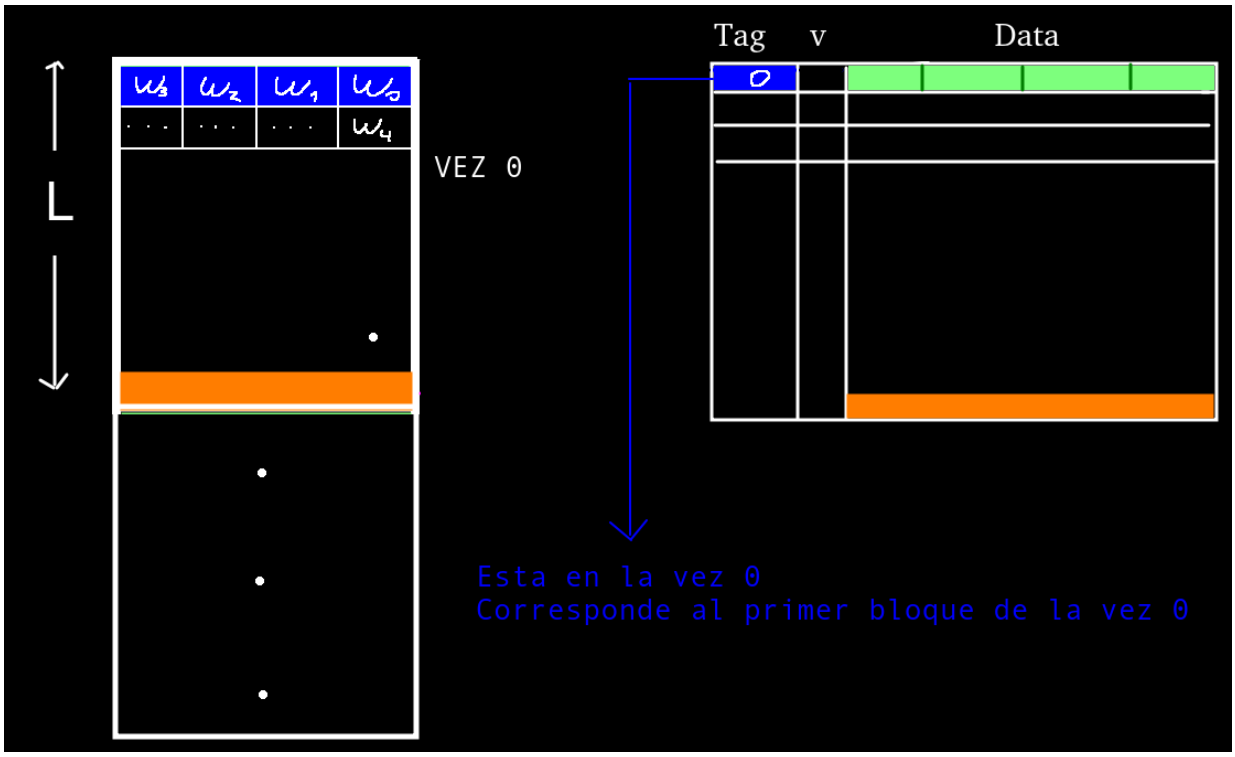
  ¿ Como se corresponden X bloques en L lineas en una de Mapeo Directo ?

L lineas dispnibles para la cache
X bloques de memoria principal
Y una representacion de cuantas veces entra X en L lineas

Explicacion



- Este criterio de correspondencia es de mapeo directo
- Sabemos que => **Bloques > Lineas de cache**

- **Veces**
  - ¿Cuantas veces entre la memoria cache en la memoria principal organizada por bloques?
  - Entonces cada vez es mapeada tal cual en la memoria cache
  - **Calculo** -> Cantidad de veces que entra la cache en X bloques de memoria principal
    - X / L (Cuantes veces entran L lineas dentro X bloques)
- **Tag**
  - Cuando nos llega un acceso a memoria este puede estar accediendo a "distintas veces de la misma"
  - Con la tag **representemos ese numero vez de la memoria**
  - **Calculo** -> Bits para direccionar N veces
    - 2^x = N veces donde x son los bits
  - 

Direccion de memoria en cache Directo(Campos)

- Offset
  - Este lo ignoro Si coinciden el tamaño en bits de Wp y Wm

        - O sea cada palabra del procesador es directamente direccionable en memoria
    - Sino necesito direccionar las Wm que estan dentro de una palabra de procesador
- Word
    - Este direcciona las Wp dentro del bloque
    - Necesito direccionar N palabras en un bloque
- Index
    - Direcciona las lineas de cache
- Tag
    - Explicado para la de mapeo directo arriba

Problema

- Si yo quiero hacer saltos consecutivos entre dos direcciones de memoria pertenecientes a distintas porciones
- Se produciria una serie de MISSES
    - Pasa a z = 2 y carga la direccion con el bloque que corresponde
    - Pero se pide en z = 1 y pues no lo tiene y carga es
    - y asi si se salta entre distintos seria **una serie de MISSES**
    - Es muy sencillo pero muy rigido ya que solo debo comparar con un solo Tag

Caches asociativas

El criterio de correspondencia de una cache asociativa permite mayor flexibilidad

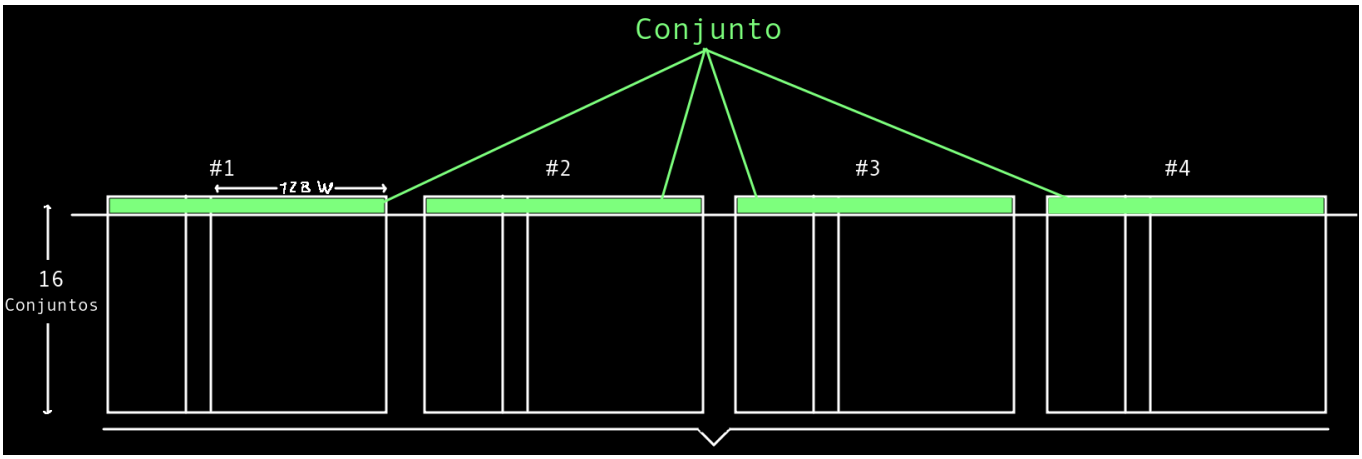
Concepto

- Cada bloque puede almacenarse en cualquier linea de cache a diferencia del directo
- Esto se logra mediante el uso de un conjunto de vias de cache

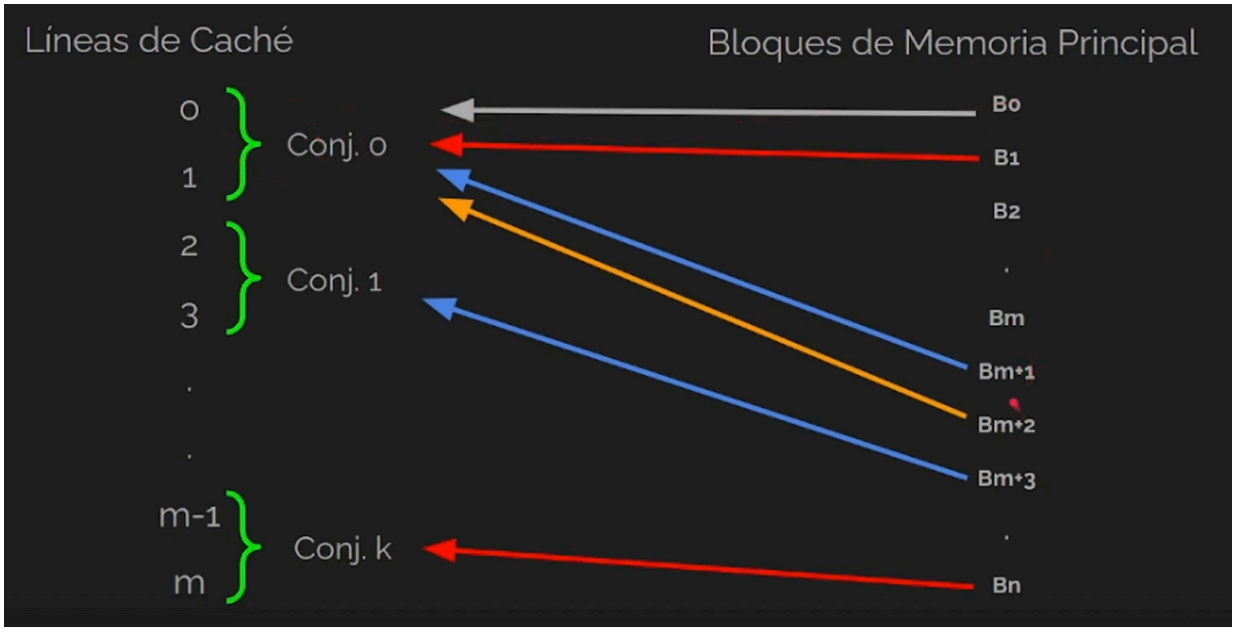
Tipos de cache asociativa

Asociativa por conjuntos

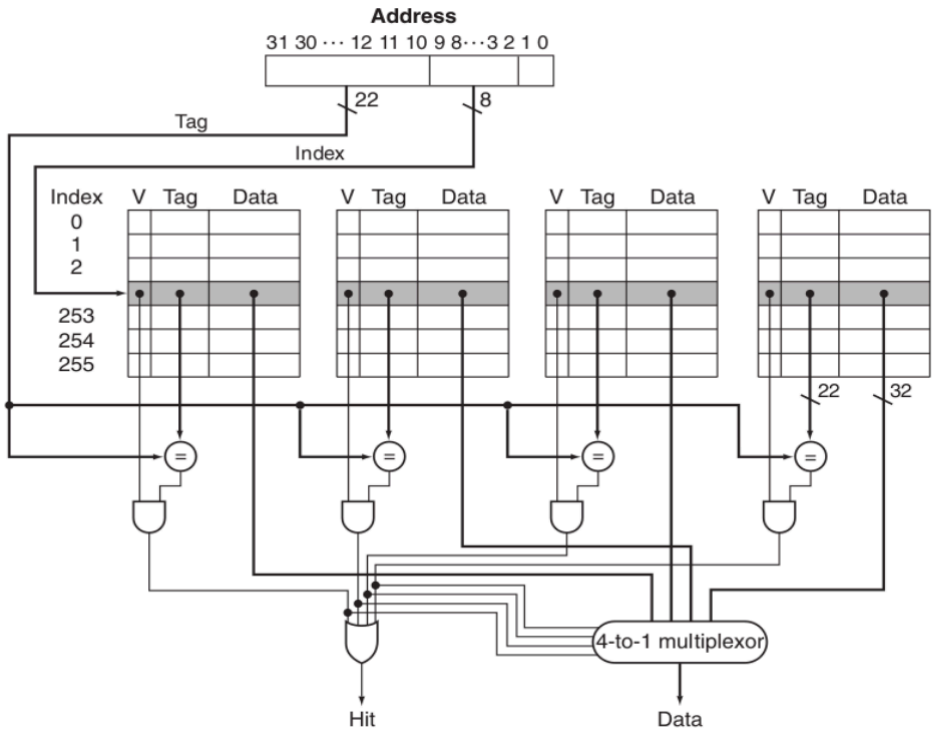
(Ejemplo de Cache de 64 lineas dividida en 4 vias)



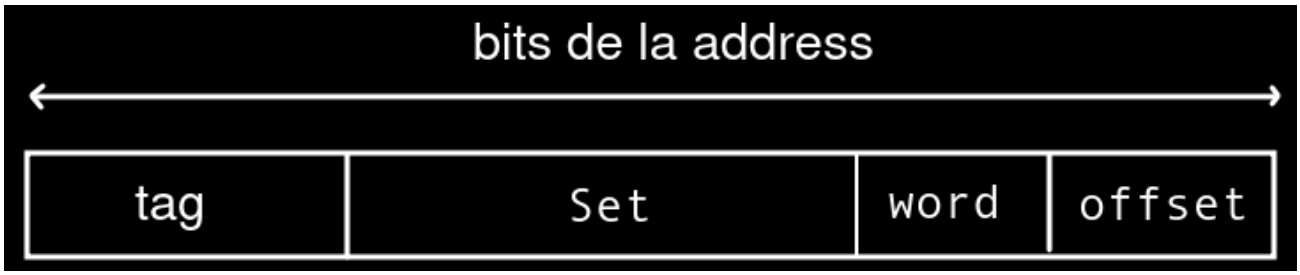
(Ejemplo de cache 2-via asociativa con bloques de una sola Wp = Wm)

- En este caso la cache se divide en varios conjuntos
- Por ende si una cache tiene N lineas estas estaran divididas en n vias
- Donde en vez de tener una linea tenemos un conjunto de Lineas (bloques)
- Un bloque dado mapea a cualquier linea en un conjunto dado
  - Un bloque B puede estar en cualquier linea del conjunto i
- La cache realiza la busqueda
  - 

    1. Revisa los bits de set para ver a que conjunto hay que buscar
    2. Revisa los bits de tag de cada conjunto para ver si oincide con el tag de la direccion
    3. Busca con los bits de word y offset la Wp

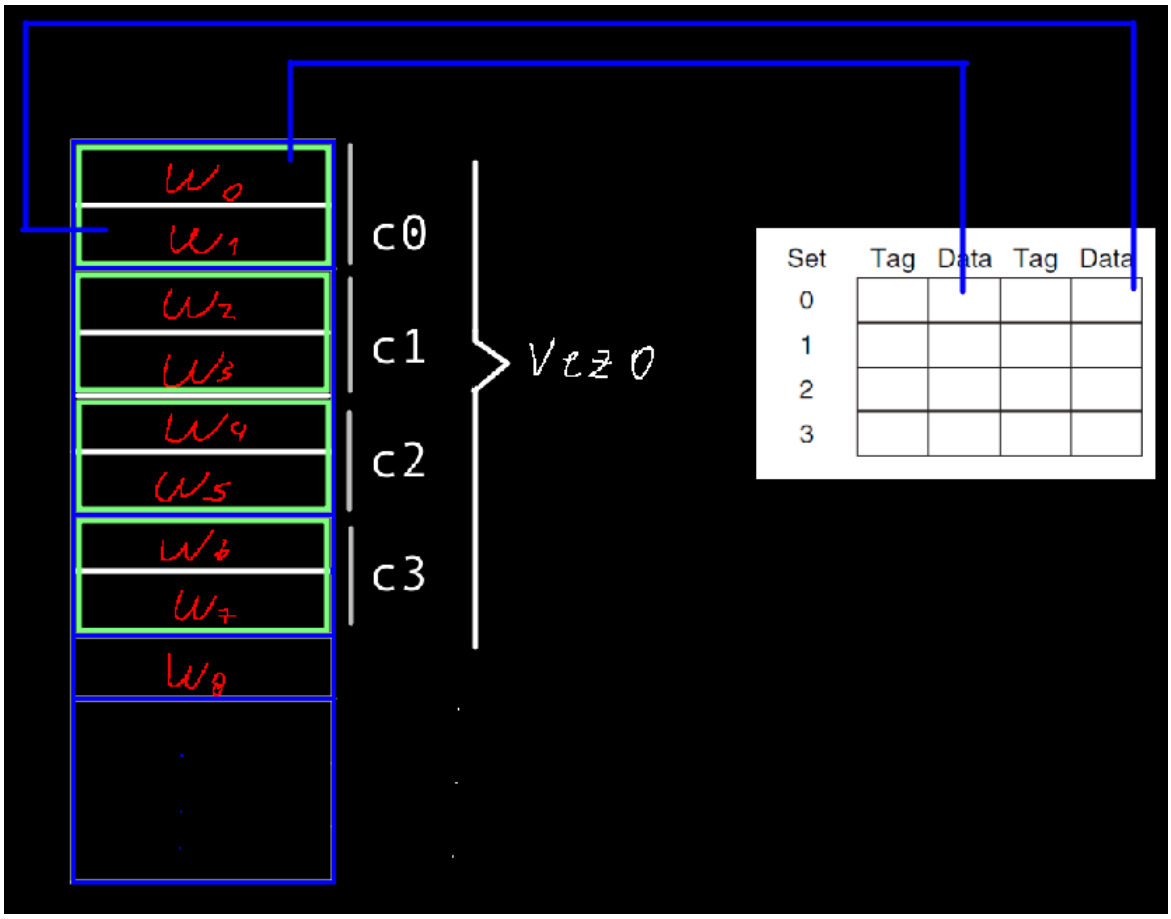


Four-way set-associative cache.



- Offset
  - Este lo ignoro Si coinciden el tamaño en bits de Wp y Wm
    - O sea cada palabra del procesador es directamente direccionable en memoria
  - Sino necesito direccionar las Wm que estan dentro de una palabra de procesador
- Word
  - Este direcciona las Wp dentro del bloque
  - Necesito direccionar N palabras en un bloque
- Set
  - Direcciona el conjunto que buscamos
  - Ojo que si me dicen que la cache tiene 32K words eso != Size(Area Data)
- Tag
  - Es lo mismo que antes pero con conjuntos
  - "Cuantas veces entra N conjuntos de cache en la memoria principal organizada por bloques"
  - O sea
  - (B/C = Cantidad de veces que entra la cache en la memoria principal organizada por bloques)

- O sea cuantos bloques pueden mapearse a cache de N conjuntos



  - Aca si nos fijamos es una palabra por bloque para 2 vias => 2 bloques por conjunto
  -

FULL-ASOCIATIVA

- Esta cache no indexo Lineas
- Hay