

Low on cache?

Assignment Project Exam Help

<https://eduassistpro.github.io/>

Add WeChat edu_assist_pro

Outline

- Memory Hierarchy
- Direct-Mapped Cache
- Types of Cache Miss
- A (long) detailed exa

Assignment Project Exam Help

<https://eduassistpro.github.io/>

Add WeChat edu_assist_pro

Memory Hierarchy (1/4)

- Processor

- executes programs

- runs on order of nan

- needs to access cod

Assignment Project Exam Help
ds

<https://eduassistpro.github.io/>

s: where are these?

- Disk

Add WeChat edu_assist_pro

- HUGE capacity (virtually limitless)

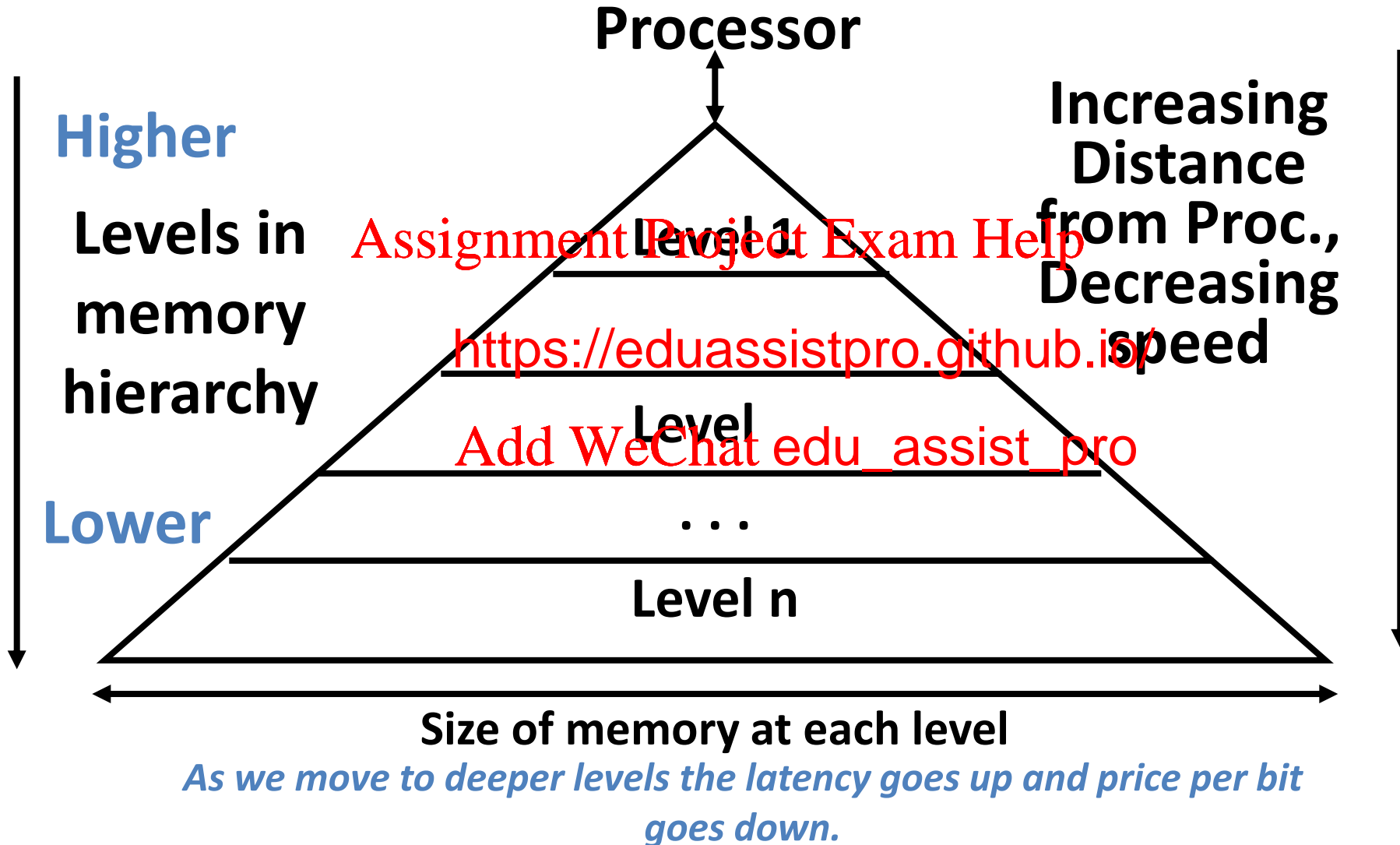
- VERY slow: runs on order of milliseconds

- so how do we account for this gap?

Memory Hierarchy (2/4)

- Memory (DRAM)
 - smaller than disk (not limitless capacity)
 - contains subset of disk portions of programs that are currently being run <https://eduassistpro.github.io/>
 - much faster than disk, memory access doesn't slow down processor quite as much
 - Problem: memory is still too slow (hundreds of nanoseconds)
 - Solution: add more layers (caches)

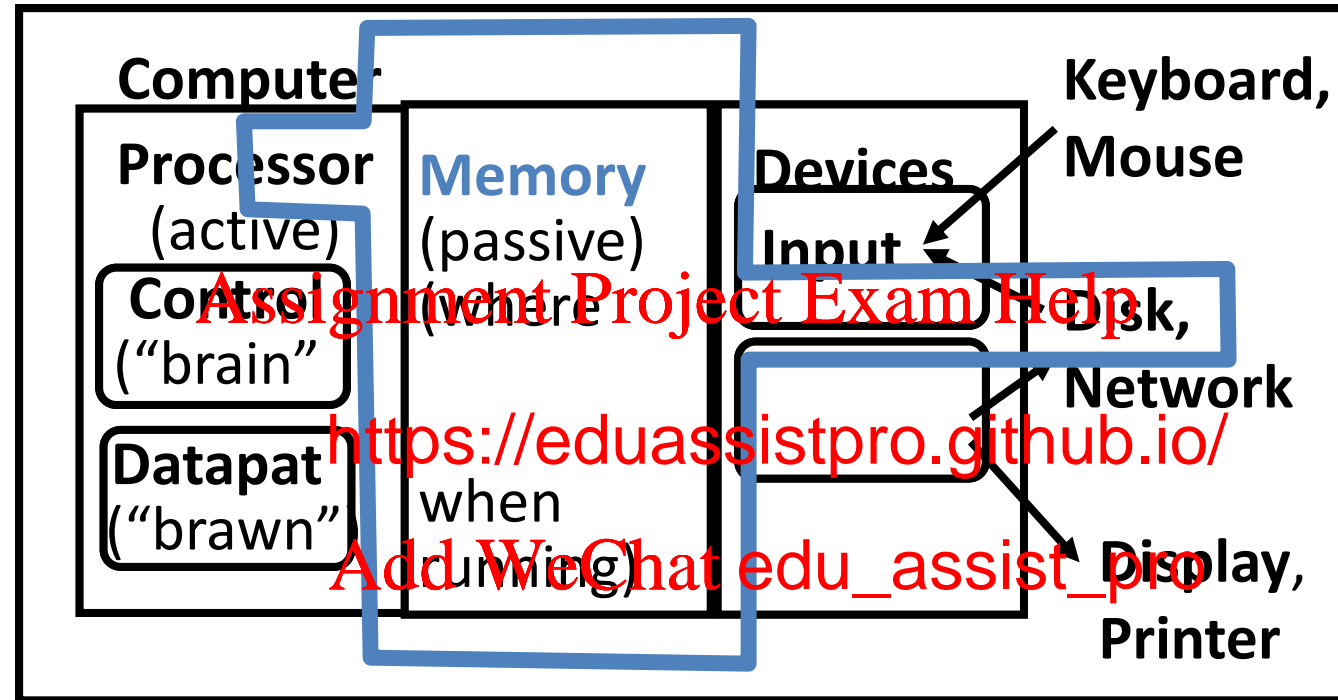
Memory Hierarchy (3/4)



Memory Hierarchy (4/4)

- If level is closer to Processor, it must...
 - Be smaller
 - Be faster
 - Contain a subset (m <https://eduassistpro.github.io/>) of lower levels beneath it (i.e., levels farther from processor)
 - Contain all the data in higher level (i.e., levels closer to processor)
- Lowest Level (usually disk) contains all available data
- Is there another level lower than disk?

Memory Hierarchy



- Purpose:
 - Faster access to large memory from processor

Memory Hierarchy Analogy: Library (1/2)

- You (the processor) are writing a term paper at a table in Schulich
- Schulich Library is e
 - essentially limitless
 - very slow to retrieve a book
- Table is memory
 - smaller capacity: means you must return book when table fills up
 - easier and faster to find a book there once you've already retrieved it

Assignment Project Exam Help

<https://eduassistpro.github.io/>

Add WeChat edu_assist_pro

Memory Hierarchy Analogy: Library (2/2)

- Open books on table are cache
 - smaller capacity: can have very few open books fit on table; again, when table fills up, you must close a book
 - much, much faster than <https://eduassistpro.github.io/>
- Illusion created: whole library on tabletop
 - Keep as many recently used books open on table as possible since likely to use again
 - Also keep as many books on table as possible, since faster than going to library

Memory Hierarchy Basis

- Disk contains everything
- When Processor needs something, bring it into to all lower levels of memory
- Cache contains copies of data that are being used
- Memory contains copies of data that are being used
- Entire idea is based on **Temporal Locality**: if we use it now, we'll want to use it again soon (a Big Idea)

Assignment Project Exam Help

<https://eduassistpro.github.io/>

Add WeChat edu_assist_pro

Athlon XP-64 Core

- The greatest share of the surface (over 50 percent) is taken up by the 1 MB L2 cache.

Assignment Project Exam Help

<https://eduassistpro.github.io/>

Add WeChat edu_assist_pro

Cache Design

- How do we organize cache?
- Where does each memory address map to?
 - Remember that caches are small, so multiple memory addresses map to the same cache line.
- How do we know which element is in the cache?
- How do we quickly locate them?

Assignment Project Exam Help

<https://eduassistpro.github.io/>

Add WeChat edu_assist_pro

Direct-Mapped Cache (1/2)

- In a *direct-mapped cache*, each memory address is associated with one possible *block* within the cache
 - Therefore, we only look at one location in the cache to see if the data exists
 - A *block* is the unit of transfer bet. cache and memory

Direct-Mapped Cache (2/2)

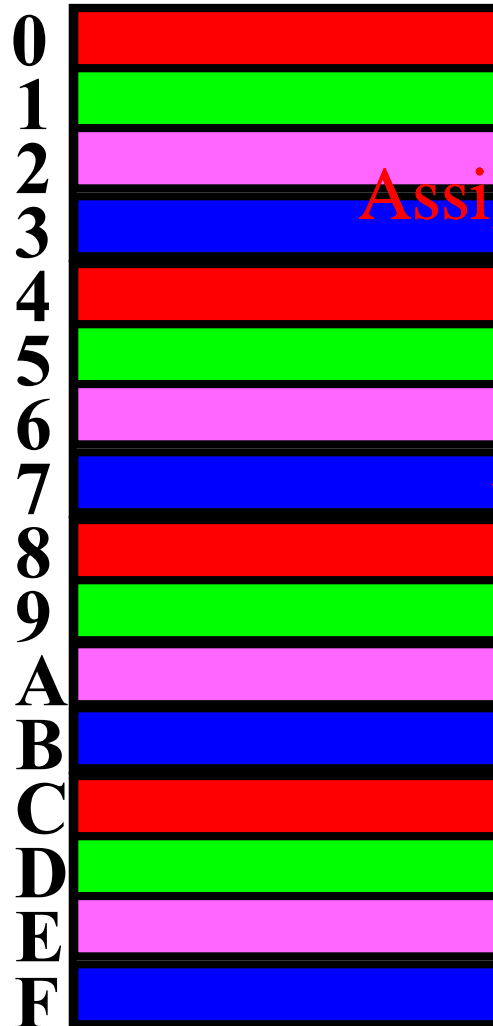
Memory

Address

Memory

Cache
Index

4 Byte Direct
Mapped Cache



Assignment Project Exam Help

<https://eduassistpro.github.io/>

Add WeChat edu_assist_pro

• C
0

tion 0 can be
data from:

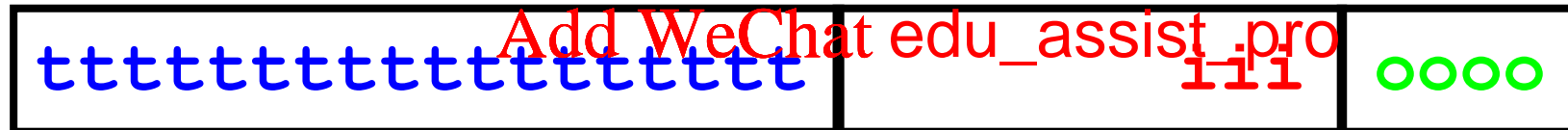
- Memory location 0, 4, 8, ...
- In general: any memory location that is multiple of 4

Issues with Direct-Mapped

1 Since multiple memory addresses map to same cache index, how do we tell which one is in there?

2 What if we have a block size > 1 byte?

- **Solution:** divide memory into blocks



tag
to check
if have
correct block

index
to
select
block

offset
byte
within
block

Direct-Mapped Cache Terminology

- All fields are read as unsigned integers.
- The **Index**: specifies the cache index (which “row” of the cache we should look in)
- The **Offset**: once we lock, specifies which byte within the block we want
- The **Tag**: the remaining bits after offset and index are determined; these are used to distinguish between all the memory addresses that map to the same location

Assignment Project Exam Help

<https://eduassistpro.github.io/>

Add WeChat edu_assist_pro

Direct-Mapped Cache Example

- Suppose we have a direct-mapped **16KB cache** with **4 word blocks**.
- Determine the size of the offset fields if we're using a 32-bit architecture

Assignment Project Exam Help

<https://eduassistpro.github.io/>

Add WeChat edu_assist_pro

Direct-Mapped Cache Example

- Offset
 - need to specify correct byte within a block
 - block contains
4 words = 16
 - need 4 bits to specify correct byte

Assignment Project Exam Help

<https://eduassistpro.github.io/>

Add WeChat edu_assist_pro

Direct-Mapped Cache Example

- Index

- need to specify correct row in cache

- cache contains 16 KB = $2^4 \cdot 2^{10} = 2^{14}$ bytes

- block contains 2^4 bytes

- # rows/cache = # blocks/cache (th lock/row)

$$\begin{aligned} &= \frac{\text{bytes/cache}}{\text{bytes/row}} \\ &= \frac{2^{14} \text{ bytes/cache}}{2^4 \text{ bytes/row}} \\ &= 2^{10} \text{ rows/cache} \end{aligned}$$

- need **10 bits** to specify this many rows

Direct-Mapped Cache Example

- Tag
 - used remaining bits as tag
 - tag length = memory fset bits minus index bits
= 32 - 4 - <https://eduassistpro.github.io/>
= 18 bits Add WeChat edu_assist_pro
 - so the tag is leftmost **18 bits** of memory address

Accessing data in a direct mapped cache

- Example: 16KB, direct-mapped, 4 word blocks
- Read 4 addresses

0x00000014
0x0000001C
0x00000034
0x00008014

- Memory values on right:
 - Let us only consider cache and memory levels of hierarchy

Memory

Address (hex)	Value of Word
...	...
00000010	a
00000014	b
00000018	c
0000001C	d
...	...
00000030	e
00000034	f
00000038	g
0000003C	h
...	...
00008010	i
00008014	j
00008018	k
0000801C	l
...	...

<https://eduassistpro.github.io/>

Add WeChat edu_assist_pro

Accessing data in a direct mapped cache

- 4 Addresses:

0x00000014, 0x0000001C, 0x00000034, 0x00008014

- 4 Addresses divided (for convenience) into

Tag, Index,

000000000

00001 0100

000000000000000000000000

0001 1100

000000000000000000000000 0000000011 0100

000000000000000000000010 0000000001 0100

Tag

Index

Offset

Assignment Project Exam Help

<https://eduassistpro.github.io/>

Add WeChat edu_assist_pro

Accessing data in a direct mapped cache

- Lets go through accessing some data in this cache
 - 16KB, direct-mapped, 4 word blocks
- Will see 3 types of events:
- cache miss: nothing in cache, so fetch from memory
- cache hit: cache block is valid and contains desired address, so read desired word
- cache miss, block replacement: wrong data is in cache at appropriate block, so discard it and fetch desired data from memory

Assignment Project Exam Help

<https://eduassistpro.github.io/>

Add WeChat edu_assist_pro

16 KB Direct Mapped Cache, 16B blocks

Valid		Example Block			
Index	Tag	0x0-3	0x4-7	0x8-b	0xc-f
0	0				
1	0	Assignment Project Exam Help			
2	0				
3	0				
4	0				
5	0				
6	0				
7	0				
...					
1022	0				
1023	0				

<https://eduassistpro.github.io/>
Valid bit: determines if anything
Add WeChat edu_assist_pro
is stored in that memory
computer initially turned on, all
entries are invalid)

Read 0x00000014

000000000000000000000000 00000000001 0100
Tag field Index field Offset

Valid					
Index	Tag	0x0-3	0x4-7	0x8-b	0xc-f
0	0				
1	0	Assignment Project Exam Help			
2	0				
3	0	https://eduassistpro.github.io/			
4	0				
5	0	Add WeChat edu_assist_pro			
6	0				
7	0				
...		...			
1022	0				
1023	0				

So we read block 1 (00000000001)

000000000000000000000000 0000000001 0100
Tag field Index field Offset

Valid					
Index	Tag	0x0-3	0x4-7	0x8-b	0xc-f
0	0				
1	0	Assignment Project Exam Help			
2	0				
3	0	https://eduassistpro.github.io/			
4	0				
5	0	Add WeChat edu_assist_pro			
6	0				
7	0				
...					
1022	0				
1023	0				

No valid data

000000000000000000000000 00000000001 0100
Tag field Index field Offset

Valid					
Index	Tag	0x0-3	0x4-7	0x8-b	0xc-f
0	0				
1	0	Assignment Project Exam Help			
2	0				
3	0	https://eduassistpro.github.io/			
4	0				
5	0	Add WeChat edu_assist_pro			
6	0				
7	0				
...					
1022	0				
1023	0				

So load that data into cache, setting tag, valid

000000000000000000000000 00000000001 0100
 Tag field Index field Offset

Valid	Index	Tag	0x0-3	0x4-7	0x8-b	0xc-f
	0	0				
	1	1	0	Assignment Project Exam Help		d
	2	0				
	3	0				
	4	0				
	5	0				
	6	0				
	7	0				
...				...		
	1022	0				
	1023	0				

Read from cache at offset, return word **b**

00000000000000000000 00000000001 0100

Tag field Index field Offset

Valid					
Index	Tag	0x0-3	0x4-7	0x8-b	0xc-f
0	0				
1	0	a	b	c	d
2	0				
3	0				
4	0				
5	0				
6	0				
7	0				
...					
1022	0				
1023	0				

Read 0x0000001C

000000000000000000000000 0000000001 **1100**
Tag field Index field Offset

Valid					
Index	Tag	0x0-3	0x4-7	0x8-b	0xc-f
0	0				
1	1	Assignment Project Exam Help			d
2	0				
3	0	https://eduassistpro.github.io/			
4	0				
5	0	Add WeChat edu_assist_pro			
6	0				
7	0				
...		...			
1022	0				
1023	0				

Data valid, tag OK, so read offset return word **d**

000000000000000000000000 00000000001 1100

Tag field

Index field

Offset

Valid					
Index	Tag	0x0-3	0x4-7	0x8-b	<u>0xc-f</u>
0	0				
<u>1</u>	<u>1</u>	Assignment Project Exam Help			<u>d</u>
2	0				
3	0				
4	0				
5	0				
6	0				
7	0				

...

...

1022	0				
1023	0				

Read 0x00000034

000000000000000000000000 0000000011 0100
Tag field Index field Offset

Valid					
Index	Tag	0x0-3	0x4-7	0x8-b	0xc-f
0	0				
1	1	Assignment Project Exam Help			
2	0				
3	0	https://eduassistpro.github.io/			
4	0				
5	0	Add WeChat edu_assist_pro			
6	0				
7	0				
...					
1022	0				
1023	0				

So read block 3

000000000000000000000000 0000000011 0100
 Tag field Index field Offset

Valid Index	Tag	0x0-3	0x4-7	0x8-b	0xc-f
0	0				
1	1	0			d
2	0				
3	0				
4	0				
5	0				
6	0				
7	0				
...			...		
1022	0				
1023	0				

No valid data

000000000000000000000000 0000000011 0100
 Tag field Index field Offset

Valid Index	Tag	0x0-3	0x4-7	0x8-b	0xc-f
0	0				
1	1	0			d
2	0				
3	0				
4	0				
5	0				
6	0				
7	0				
...			...		
1022	0				
1023	0				

Assignment Project Exam Help

<https://eduassistpro.github.io/>

Add WeChat edu_assist_pro

Load that cache block, return word f

000000000000000000000000 0000000011 0100
 Tag field Index field Offset

Valid	Index	Tag	0x0-3	0x4-7	0x8-b	0xc-f
	0	0				
	1	1	0			d
	2	0				
	3	1	0			h
	4	0				
	5	0				
	6	0				
	7	0				
...	...					
1022	0					
1023	0					

Read 0x00008014

0000000000000000000010 0000000001 0100
Tag field Index field Offset

Valid					
Index	Tag	0x0-3	0x4-7	0x8-b	0xc-f
0	0				
1	1	Assignment Project Exam Help			d
2	0				
3	1	https://eduassistpro.github.io/			h
4	0				
5	0	Add WeChat edu_assist_pro			
6	0				
7	0				
...		...			
1022	0				
1023	0				

So we read block 1, Data is Valid

0000000000000000000010 0000000001 0100
Tag field Index field Offset

Valid

Index	Tag	0x0-3	0x4-7	0x8-b	0xc-f
0	0				
1	1	0	Assignment Project Exam Help		d
2	0				
3	1	0	https://eduassistpro.github.io/		h
4	0				
5	0		Add WeChat edu_assist_pro		
6	0				
7	0				
...			...		
1022	0				
1023	0				

Cache Block 1 Tag does not match (0 != 2)

0000000000000000000010												000000000001												0100			
Tag field												Index field												Offset			
Valid		Tag		0x0-3				0x4-7				0x8-b				0xc-f											
Index																											
0	0																										
1	1	0	Assignment Project Exam Help																d								
2	0																										
3	1	0	https://eduassistpro.github.io/																h								
4	0																										
5	0		Add WeChat edu_assist_pro																								
6	0																										
7	0																										
...																											
1022	0																										
1023	0																										

Miss, replace block 1 with new data & tag

0000000000000000000010 00000000001 0100
Tag field Index field Offset

Valid					
Index	Tag	0x0-3	0x4-7	0x8-b	0xc-f
0	0				
1	1	2	Assignment Project Exam Help		l
2	0				
3	1	0	https://eduassistpro.github.io/		h
4	0				
5	0		Add WeChat edu_assist_pro		
6	0				
7	0				
...		...			
1022	0				
1023	0				

And return word j

0000000000000000000010 00000000001 0100
Tag field Index field Offset

Valid					
Index	Tag	0x0-3	0x4-7	0x8-b	0xc-f
0	0				
1	1	2	Assignment Project Exam Help		l
2	0				
3	1	0	https://eduassistpro.github.io/		h
4	0				
5	0		Add WeChat edu_assist_pro		
6	0				
7	0				
...		...			
1022	0				
1023	0				

Do an example yourself. What happens?

- Cache: Hit, Miss, Miss with replace ?
Values returned: a ,b, c, d, e, ..., k, l ?
- Read address 0x00000030 ?
00000000000000000000 0000000011 0000
- Read address 0x0000001c ?
00000000000000000000 0000000001 1100

Cache

Index	Valid	Tag	0	https://eduassistpro.github.io/	x8-b	0xc-f
0	0			Add WeChat edu_assist_pro		
1	1	2	i	j	k	l
2	0					
3	1	0	e	f	g	h
4	0					
5	0					
6	0					
7	0					
...				...		

Answers

Memory Address Value of Word

- 0x00000030 a hit
Index = 3, Tag matches,
Offset = 0, value = e

...	...
00000010	a
00000014	b
00000018	c
0000001c	d

- 0x0000001c
Index = 1, Tag
replace from memory,
Offset = 0xc, value = d

...	...
00000030	e
00000034	f
00000038	g
0000003c	h

- Therefore, returned values are:
 - 0x00000030 = e
 - 0x0000001c = d

...	...
00008010	i
00008014	j
00008018	k
0000801c	l
...	...

“And in Conclusion...”

- We would like to have the capacity of disk at the speed of the processor: unfortunately this is not feasible.
- So we create a memory hierarchy:
 - each successively higher level caches data from next lower level
 - exploits temporal locality and spatial locality
 - do the common case fast, worry less about the exceptions (design principle of MIPS)
- Locality of reference is a Big Idea

Review and More Information

- Sections 5.1 - 5.3 of textbook

Assignment Project Exam Help

<https://eduassistpro.github.io/>

Add WeChat edu_assist_pro