Data cache & Locality Pipelining

Recap

Philip Shun Jensen

UNIVERSITY OF COPENHAGEN

Agenda

- Data Cache & Locality theory
  - Memory Hierarchy
  - Array layouts
  - Locality
  - Cache Organisation
  - Exam Questions
- Pipelining theory
  - Resources
  - Data Hazards
  - Full Forwarding
  - Branch Prediction
  - Exam Questions

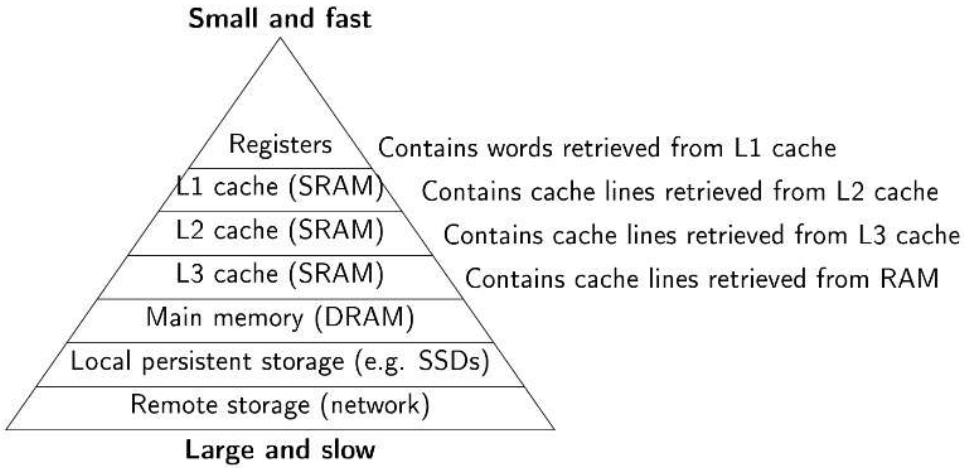
Data cache & Locality

Recap

UNIVERSITY OF COPENHAGEN

The Memory Hierarchy

- Cache definition
  - A smaller, faster storage device that acts as a staging area for the subset of the data in a larger, slower device
- The idea
  - The smaller and faster device at level k acts as a cache for the larger and slower device at level k+1

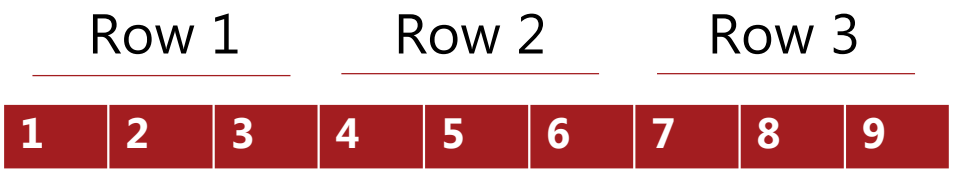


From slide 5 of "Dynamic Memory and Cache"

Array Layouts

1	2	3
4	5	6
7	8	9

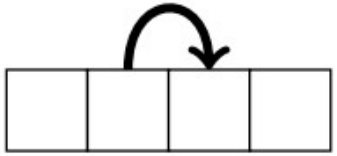
- Row-major order (used in C)



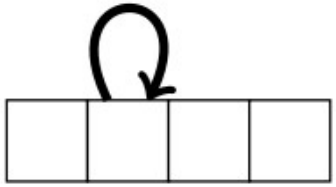
- Column-major order (used in MatLab)

Locality

- Spatial locality
  - Accessing data that is close to the data that was recently accessed



- Temporal locality
  - Accessing data that was recently accessed



- Stride
  - How large are the jumps between memory accesses?

Exercise 5.1, COD

```
for (I=0; I<8; I++)
  for (J=0; J<8000; J++)
    A[I][J]=B[I][0]+A[J][I];
```

C code

- Which variable reference(s) exhibit temporal locality?

Exercise 5.1, COD

```
for (I=0; I<8; I++)
  for (J=0; J<8000; J++)
    A[I][J]=B[I][0]+A[J][I];
```
C code

- Which variable reference(s) exhibit temporal locality?
  - I, J, B[I][0]
  - I remains constant each time J runs
  - J is accessed more times in each iteration of the inner loop
  - B[I][0] is accessed 8000 time within each I loop

Exercise 5.1, COD

```
for (I=0; I<8; I++)
  for (J=0; J<8000; J++)
    A[I][J]=B[I][0]+A[J][I];
```
C code

- Which variable reference(s) exhibit temporal locality?
  - I, J, B[I][0]
  - I remains constant each time J runs
  - J is accessed more times in each iteration of the inner loop
  - B[I][0] is accessed 8000 time within each I loop
- Which variable reference(s) exhibit spatial locality?

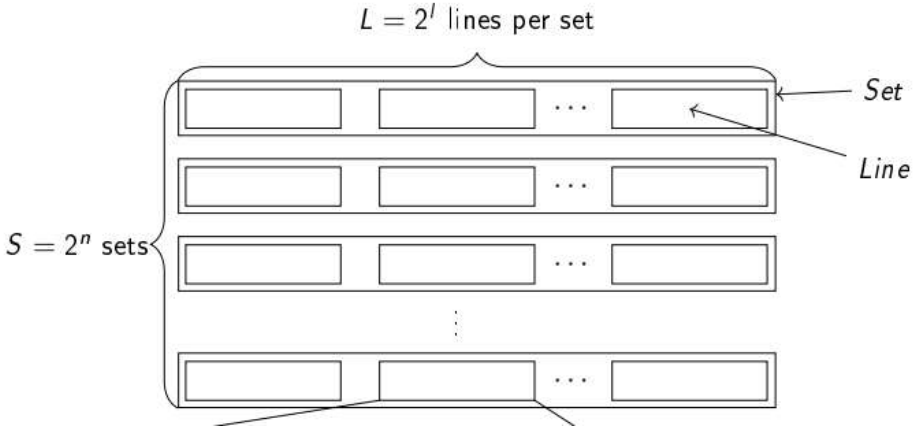
Exercise 5.1, COD

```
for (I=0; I<8; I++)
  for (J=0; J<8000; J++)
    A[I][J]=B[I][0]+A[J][I];
```
C code

- Which variable reference(s) exhibit temporal locality?
  - I, J, B[I][0]
  - I remains constant each time J runs
  - J is accessed more times in each iteration of the inner loop
  - B[I][0] is accessed 8000 time within each I loop
- Which variable reference(s) exhibit spatial locality?
  - A[I][J]
  - As J increases in the loop, (A[I][0], A[I][1], ⋯, A[I][J]) the elements are accessed consecutively in the same row for A

Cache organisation

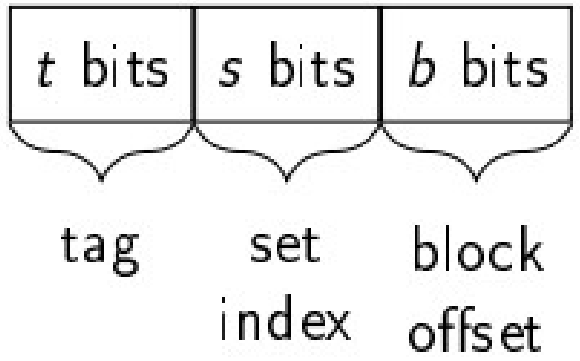
- N-way set associative
  - The are N block in each set
- Direct mapped (one-way set associative)
  - One block for each set
  - Each memory location is mapped to one location in the cache
- Fully associative
  - Cache is one set
  - Memory location can be mapped to anywhere in the cache
  - If the cache has space for 8 blocks, and it is fully associative, it would be 8-way set associative



From slide 25 of "Dynamic Memory and Cache"

Address translation

- Order in bits (left to right)
  - Tag, Set index, Block offset
- Offset bits (b)
  - How many bits needed to represent a block
  - $b = \log_2(blocksizeInBytes)$
- How many sets are in n-way associative cache?
  - $setCount = \frac{cachesizeInBytes}{n \cdot blocksizeInBytes}$
- Set index bits (s)
  - How many bits needed to represent the sets
  - $s = \log_2(setCount)$
- Tag (t)
  - The rest
  - $t = bitsInAddress - (s + b)$



From slide 28 of "Dynamic Memory and Cache"

Unit conversion

- Memory is byte-addressed, so it is a good idea to convert everything to bytes
  - Kibibyte (KiB) = 1024 bytes
  - Mebibyte (MiB) = 1024 KiB
  - Kilobit = 125 bytes

Exam Question 2024/25 Re-Exam

- READ THE TEXT CAREFULLY!

1.1 Data Cache and Locality (about 10 %)

A byte-addressed machine with 32-bit addresses is equipped with a 8 kibibyte 2-way set-associative cache with a block size of 16 bytes. The cache uses "least-recently-used replacement policy".

- bitsInAddress = 32
- cacheSize = 8KiB
- n = 2
- blockSize = 16 bytes
- LRU (important later)

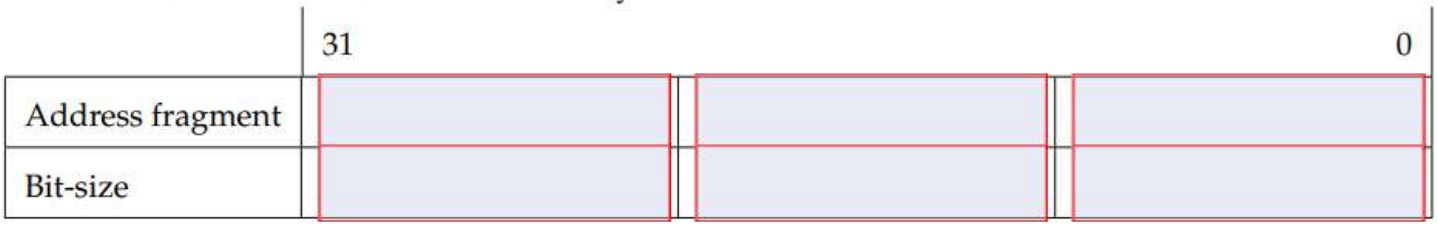
Exam Question 2024/25 Re-Exam

bitsInAddress = 32
cacheSize = 8KiB
n = 2
blockSize = 16 bytes

Question 1.1.1: The address is separated into the following fragments when the cache is accessed

- block offset,
- cache tag, and
- set index.

What is bit-size of each and how are they ordered?



Exam Question 2024/25 Re-Exam

bitsInAddress = 32
cacheSize = 8KiB
n = 2
blockSize = 16 bytes

Question 1.1.1: The address is separated into the following fragments when the cache is accessed

- block offset,
- cache tag, and
- set index.

What is bit-size of each and how are they ordered?

	31		0
Address fragment	Cache tag	Set index	Block offset
Bit-size			

Exam Question 2024/25 Re-Exam

bitsInAddress = 32
cacheSize = 8KiB
n = 2
blockSize = 16 bytes

Question 1.1.1: The address is separated into the following fragments when the cache is accessed

- block offset,
- cache tag, and
- set index.

What is bit-size of each and how are they ordered?

	31		0
Address fragment	Cache tag	Set index	Block offset
Bit-size			4

$b = \log_2(blocksizeInBytes)$
$b = \log_2 16 = 4$

Exam Question 2024/25 Re-Exam

bitsInAddress = 32
cacheSize = 8KiB
n = 2
blockSize = 16 bytes

Question 1.1.1: The address is separated into the following fragments when the cache is accessed

- block offset,
- cache tag, and
- set index.

What is bit-size of each and how are they ordered?

	31		0
Address fragment	Cache tag	Set index	Block offset
Bit-size		8	4

$cachesizeInBytes = 8KiB \cdot 1024 = 8192$

$setCount = \frac{cachesizeInBytes}{n \cdot blocksizeInBytes}$

$setCount = \frac{8192}{2 \cdot 16} = 256$

$s = \log_2 256 = 8$

$b = \log_2(blocksizeInBytes)$

$b = \log_2 16 = 4$

Exam Question 2024/25 Re-Exam

bitsInAddress = 32
cacheSize = 8KiB
n = 2
blockSize = 16 bytes

Question 1.1.1: The address is separated into the following fragments when the cache is accessed

- block offset,
- cache tag, and
- set index.

What is bit-size of each and how are they ordered?

	31		0
Address fragment	Cache tag	Set index	Block offset
Bit-size	20	8	4

$t = bitsInAddress - (s + b)$
$t = 32 - (8 + 4) = 20$

$cachesizeInBytes = 8KiB \cdot 1024$
$= 8192$
$setCount = \frac{cachesizeInBytes}{n \cdot blocksizeInBytes}$
$setCount = \frac{8192}{2 \cdot 16} = 256$
$s = \log_2 256 = 8$

$b = \log_2(blocksizeInBytes)$
$b = \log_2 16 = 4$

Exam Question 2024/25 Re-Exam

Question 1.1.2:

Calculate for the following addresses the value of block offset, cache tag, and set index.

Address: 0x76543210

(Maximum 5 lines.)

bitsInAddress = 32
cacheSize = 8KiB
n = 2
blockSize = 16 bytes
Block offset = 4 bits
Set index = 8 bits
Cache tag = 20 bits

Exam Question 2024/25 Re-Exam

Question 1.1.2:
Calculate for the following addresses the value of block offset, cache tag, and set index.
Address: 0x76543210

(Maximum 5 lines.)

0111 0110 0101 0100 0011 0010 0001 0000

bitsInAddress = 32
cacheSize = 8KiB
n = 2
blockSize = 16 bytes
Block offset = 4 bits
Set index = 8 bits
Cache tag = 20 bits

Exam Question 2024/25 Re-Exam

Question 1.1.2:

Calculate for the following addresses the value of block offset, cache tag, and set index.

Address: 0x76543210

(Maximum 5 lines.)

0111 0110 0101 0100 0011	0010 0001	0000
Cache tag	Set index	Block offset

bitsInAddress = 32
cacheSize = 8KiB
n = 2
blockSize = 16 bytes
Block offset = 4 bits
Set index = 8 bits
Cache tag = 20 bits

Exam Question 2024/25 Re-Exam

bitsInAddress = 32
cacheSize = 8KiB
n = 2
blockSize = 16 bytes
Block offset = 4 bits
Set index = 8 bits
Cache tag = 20 bits

Question 1.1.2:
Calculate for the following addresses the value of block offset, cache tag, and set index.
Address: 0x76543210

Block offset: 0x0
Set index: 0x21
Cache tag: 076543

(Maximum 5 lines.)

0111 0110 0101 0100 0011	0010 0001	0000
Cache tag	Set index	Block offset

Exam Question 2024/25 Re-Exam

bitsInAddress = 32
cacheSize = 8KiB
n = 2
blockSize = 16 bytes
Block offset = 4 bits
Set index = 8 bits
Cache tag = 20 bits

Question 1.1.3: On the machine, the following stream of cache accesses are performed (read from top to bottom). Indicate for each of the address references:

- the set index,
- if the cache access is a miss or a hit, and
- the cache tags in LRU-order (Least recently used tag last) that are in the affected set after the access.

Addresses are given in hexa-decimal notation. Assume the cache is cold on entry.

- LRU (Least Recently Used): The block that is replaced in the set, is the one that has been unused for the longest
  - Find set index of the address
  - Search for the block in the set by comparing tags
  - If found: cache hit, and update the block such it is visible that it was just used
  - Not found: cache miss,
    - If one or more blocks are not valid (check valid bit), replace one of these with the new block
    - If all blocks in the set are valid, replace the least recently used block, and update that new block such it is visible that it was just used

Exam Question 2024/25 Re-Exam

Reference	Set index	Hit/Miss	State of Tags
0x0100004			
0x010000C			
0x0010008			
0x0110004			
0x0010008			
0x0300004			
0x0300010			
0x0010008			
0x0410004			
0x0A00008			
0x0110004			
0x0A00000			
0x0A00014			
0x0300010			

bitsInAddress = 32
cacheSize = 8KiB
n = 2
blockSize = 16 bytes
Block offset = 4 bits
Set index = 8 bits
Cache tag = 20 bits
Order = LRU
Cache = Cold

Exam Question 2024/25 Re-Exam

Reference	Set index	Hit/Miss	State of Tags
0x0100004	0x0	Miss	0x100
0x010000C			
0x0010008			
0x0110004			
0x0010008			
0x0300004			
0x0300010			
0x0010008			
0x0410004			
0x0A00008			
0x0110004			
0x0A00000			
0x0A00014			
0x0300010			

bitsInAddress = 32
cacheSize = 8KiB
n = 2
blockSize = 16 bytes
Block offset = 4 bits
Set index = 8 bits
Cache tag = 20 bits
Order = LRU
Cache = Cold

0x0100004=

0000 0000 0001 0000 0000	0000 0000	0100
0x100	0x0	

Exam Question 2024/25 Re-Exam

Reference	Set index	Hit/Miss	State of Tags
0x0100004	0x0	Miss	0x100
0x010000C	0x0	Hit	0x100
0x0010008			
0x0110004			
0x0010008			
0x0300004			
0x0300010			
0x0010008			
0x0410004			
0x0A00008			
0x0110004			
0x0A00000			
0x0A00014			
0x0300010			

bitsInAddress = 32
cacheSize = 8KiB
n = 2
blockSize = 16 bytes
Block offset = 4 bits
Set index = 8 bits
Cache tag = 20 bits
Order = LRU
Cache = Cold

0x010000C=

0000 0000 0001 0000 0000	0000 0000	1100
0x100	0x0	

Exam Question 2024/25 Re-Exam

Reference	Set index	Hit/Miss	State of Tags
0x0100004	0x0	Miss	0x100
0x010000C	0x0	Hit	0x100
0x0010008	0x0	Miss	0x010,0x100
0x0110004			
0x0010008			
0x0300004			
0x0300010			
0x0010008			
0x0410004			
0x0A00008			
0x0110004			
0x0A00000			
0x0A00014			
0x0300010			

bitsInAddress = 32
cacheSize = 8KiB
n = 2
blockSize = 16 bytes
Block offset = 4 bits
Set index = 8 bits
Cache tag = 20 bits
Order = LRU
Cache = Cold

0x0010008=

0000 0000 0000 0001 0000	0000 0000	1000
0x010	0x0	

Exam Question 2024/25 Re-Exam

Reference	Set index	Hit/Miss	State of Tags
0x0100004	0x0	Miss	0x100
0x010000C	0x0	Hit	0x100
0x0010008	0x0	Miss	0x010,0x100
0x0110004	0x0	Miss	0x110,0x010
0x0010008			
0x0300004			
0x0300010			
0x0010008			
0x0410004			
0x0A00008			
0x0110004			
0x0A00000			
0x0A00014			
0x0300010			

bitsInAddress = 32
cacheSize = 8KiB
n = 2
blockSize = 16 bytes
Block offset = 4 bits
Set index = 8 bits
Cache tag = 20 bits
Order = LRU
Cache = Cold

0x0110004=

0000 0000 0001 0001 0000	0000 0000	0100
0x110	0x0	

Exam Question 2024/25 Re-Exam

Reference	Set index	Hit/Miss	State of Tags
0x0100004	0x0	Miss	0x100
0x010000C	0x0	Hit	0x100
0x0010008	0x0	Miss	0x010,0x100
0x0110004	0x0	Miss	0x110,0x010
0x0010008	0x0	Hit	0x010,0x110
0x0300004			
0x0300010			
0x0010008			
0x0410004			
0x0A00008			
0x0110004			
0x0A00000			
0x0A00014			
0x0300010			

bitsInAddress = 32
cacheSize = 8KiB
n = 2
blockSize = 16 bytes
Block offset = 4 bits
Set index = 8 bits
Cache tag = 20 bits
Order = LRU
Cache = Cold

0x0010008=

0000 0000 0000 0001 0000	0000 0000	1000
0x010	0x0	

Exam Question 2024/25 Re-Exam

Reference	Set index	Hit/Miss	State of Tags
0x0100004	0x0	Miss	0x100
0x010000C	0x0	Hit	0x100
0x0010008	0x0	Miss	0x010,0x100
0x0110004	0x0	Miss	0x110,0x010
0x0010008	0x0	Hit	0x010,0x110
0x0300004	0x0	Miss	0x300,0x010
0x0300010			
0x0010008			
0x0410004			
0x0A00008			
0x0110004			
0x0A00000			
0x0A00014			
0x0300010			

bitsInAddress = 32
cacheSize = 8KiB
n = 2
blockSize = 16 bytes
Block offset = 4 bits
Set index = 8 bits
Cache tag = 20 bits
Order = LRU
Cache = Cold

0x0300004=

0000 0000 0011 0000 0000	0000 0000	0100
0x300	0x0	

Exam Question 2024/25 Re-Exam

Reference	Set index	Hit/Miss	State of Tags
0x0100004	0x0	Miss	0x100
0x010000C	0x0	Hit	0x100
0x0010008	0x0	Miss	0x010,0x100
0x0110004	0x0	Miss	0x110,0x010
0x0010008	0x0	Hit	0x010,0x110
0x0300004	0x0	Miss	0x300,0x010
0x0300010	0x1	Miss	0x300
0x0010008			
0x0410004			
0x0A00008			
0x0110004			
0x0A00000			
0x0A00014			
0x0300010			

bitsInAddress = 32
cacheSize = 8KiB
n = 2
blockSize = 16 bytes
Block offset = 4 bits
Set index = 8 bits
Cache tag = 20 bits
Order = LRU
Cache = Cold

0x0300010=

0000 0000 0011 0000 0000	0000 0001	0000
0x300	0x1	

Exam Question 2024/25 Re-Exam

Reference	Set index	Hit/Miss	State of Tags
0x0100004	0x0	Miss	0x100
0x010000C	0x0	Hit	0x100
0x0010008	0x0	Miss	0x010,0x100
0x0110004	0x0	Miss	0x110,0x010
0x0010008	0x0	Hit	0x010,0x110
0x0300004	0x0	Miss	0x300,0x010
0x0300010	0x1	Miss	0x300
0x0010008	0x0	Hit	0x010,0x300
0x0410004			
0x0A00008			
0x0110004			
0x0A00000			
0x0A00014			
0x0300010			

bitsInAddress = 32
cacheSize = 8KiB
n = 2
blockSize = 16 bytes
Block offset = 4 bits
Set index = 8 bits
Cache tag = 20 bits
Order = LRU
Cache = Cold

0x0010008=

0000 0000 0000 0001 0000	0000 0000	1000
0x010	0x0	

Exam Question 2024/25 Re-Exam

Reference	Set index	Hit/Miss	State of Tags
0x0100004	0x0	Miss	0x100
0x010000C	0x0	Hit	0x100
0x0010008	0x0	Miss	0x010,0x100
0x0110004	0x0	Miss	0x110,0x010
0x0010008	0x0	Hit	0x010,0x110
0x0300004	0x0	Miss	0x300,0x010
0x0300010	0x1	Miss	0x300
0x0010008	0x0	Hit	0x010,0x300
0x0410004	0x0	Miss	0x410,0x010
0x0A00008	0x0	Miss	0xA00,0x410
0x0110004	0x0	Miss	0x110,0xA00
0x0A00000	0x0	Hit	0xA00,0x110
0x0A00014	0x1	Miss	0xA00,0x300
0x0300010			

bitsInAddress = 32
cacheSize = 8KiB
n = 2
blockSize = 16 bytes
Block offset = 4 bits
Set index = 8 bits
Cache tag = 20 bits
Order = LRU
Cache = Cold

0x0A00014=

0000 0000 1010 0000 0000	0000 0001	0100
0xA00	0x1	

Exam Question 2024/25 Re-Exam

Reference	Set index	Hit/Miss	State of Tags
0x0100004	0x0	Miss	0x100
0x010000C	0x0	Hit	0x100
0x0010008	0x0	Miss	0x010,0x100
0x0110004	0x0	Miss	0x110,0x010
0x0010008	0x0	Hit	0x010,0x110
0x0300004	0x0	Miss	0x300,0x010
0x0300010	0x1	Miss	0x300
0x0010008	0x0	Hit	0x010,0x300
0x0410004	0x0	Miss	0x410,0x010
0x0A00008	0x0	Miss	0xA00,0x410
0x0110004	0x0	Miss	0x110,0xA00
0x0A00000	0x0	Hit	0xA00,0x110
0x0A00014	0x1	Miss	0xA00,0x300
0x0300010	0x1	Hit	0x300,0xA00

bitsInAddress = 32
cacheSize = 8KiB
n = 2
blockSize = 16 bytes
Block offset = 4 bits
Set index = 8 bits
Cache tag = 20 bits
Order = LRU
Cache = Cold

0x0300010=

0000 0000 0011 0000 0000	0000 0001	0000
0x030	0x1	

Pipelining

Recap

UNIVERSITY OF COPENHAGEN

Resources

- Simple 5-stage pipeline
  - Fetch
  - Decode
  - Execute
  - Memory (load or store)
  - Writeback

All:	Fe	De	Ex	Me	Wb

- Fe: 1, De: 1, Ex: 1, Me: 1, Wb: 1
  - This means we have one of each resource, which means only one instruction can be in the same phase at a time

Resources

- 9-stage pipeline realistic cache access
  - Fetch
  - Decode
  - Execute
  - Memory (load or store)
  - Writeback

All:	Fa	Fb	Fc	De	Ex	Ma	Mb	Mc	Wb

- Fe: 1, De: 1, Ex: 1, Me: 1, Wb: 1
  - This means we have one of each resource, which means only one instruction can be in the same phase at a time

Resources

- 2-way in-order Superscalar
  - Fetch
  - Decode
  - Execute
  - Address Generation
  - Memory (load or store)
  - Writeback

Load	Fe	De	Ag	Me	Wb
Store	Fe	De	Ag	Me	
Other	Fe	De	Ex	Wb	
Branch	Fe	De	Ex		

- Fe: 2, De: 2, Ex: 2, Ag: 1, Me: 1, Wb: 2
  - We only have one Ag and one Me, since we only allow one cache-access pr. clock cycle.
  - Only one instruction pr register from Ex and forwards

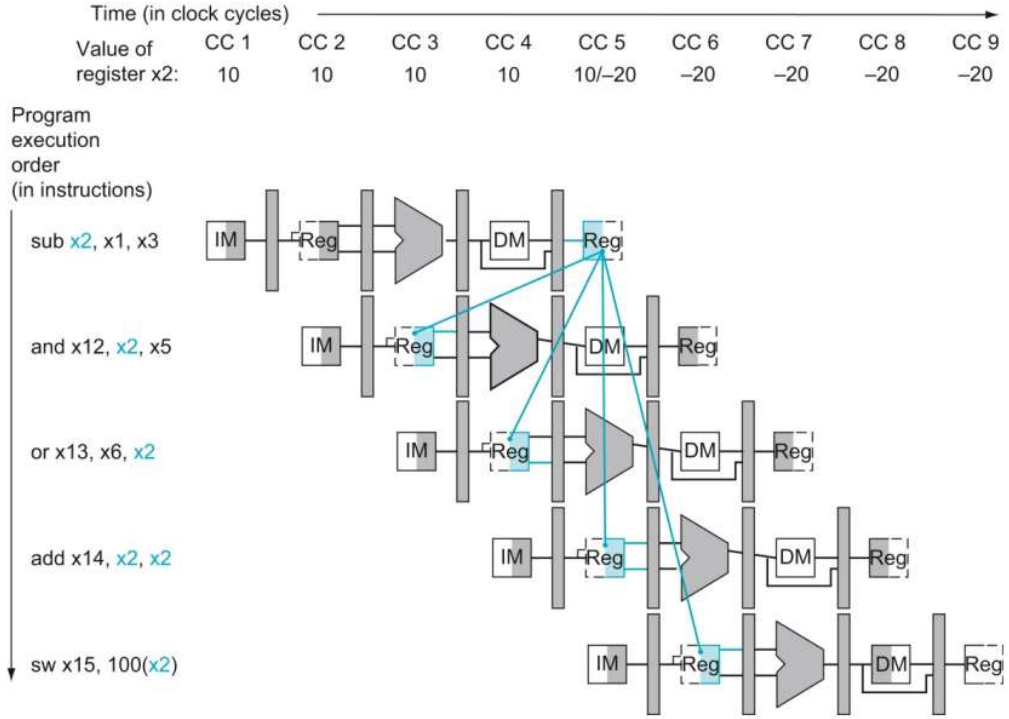
Resources

- 4-way out-of-order realistic cache access
  - Fa, Fb, Fc, De, Fu, Al, Rn, Ca, Cb: 4
  - [Qu-Ca]: 64
  - ALU-op: 1 clock cycle latency
  - Load-op: 4 clock cycles latency
  - Xx/xx = In-order/out-of-order

Load	Fa	Fb	Fc	De	Fu	Al	Rn	Qu	pk	rd	ag	ma	mb	mc	wb	Ca	Cb
Store	Fa	Fb	Fc	De	Fu	Al	Rn	Qu	pk	rd	ag	ma	mb	mc	Ca	Cb	
Other	Fa	Fb	Fc	De	Fu	Al	Rn	Qu	pk	rd	ex	wb	Ca	Cb			
Branch	Fa	Fb	Fc	De	Fu	Al	Rn	Qu	pk	rd	ex	Ca	Cb				

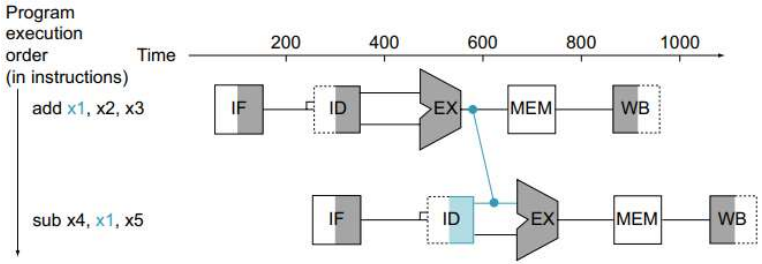
Data Hazards

- When an instruction depends on the result from a previous instruction, which hasn't been computed yet
- Fixed by stalling (expensive) and/or forwarding

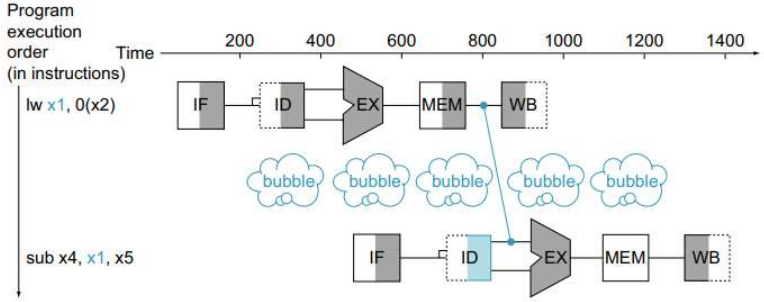


Forwarding

- Alu operations
  - Result gets forwarded from the Ex stage



- Load operations
  - Result gets forwarded from the Me stage



Branch Prediction

- Backward branch (loop) taken
  - Predicted at De stage
  - PC gets written after the decode stage
- Backward branch not taken
  - Predicted at Ex stage
  - PC gets written afther the execute stage
- Forward branch (if/else) taken
  - Predicted at Ex stage
  - PC gets written after the execute stage
- If we guess wrong we lose cycles

Exam Question 2024/25 Re-Exam

1.2 Microarchitecture and execution diagram (about 10 %)

Consider the following fragment of a program execution (2 iterations of an inner loop)

```
1  .L3:    lhu     a5,0(a1)
2          addi    a1,a1,2
3          slli    a5,a5,2
4          add     a5,a2,a5
5          lw      a5,0(a5)
6          add     a0,a0,a5
7          bne     a4,a1,.L3
8  .L3:    lhu     a5,0(a1)
9          addi    a1,a1,2
10         slli    a5,a5,2
11         add     a5,a2,a5
12         lw      a5,0(a5)
13         add     a0,a0,a5
14         bne     a4,a1,.L3
```

All questions in this exercise refer to how this instruction sequence is executed on different micro-architectures.

Exam Question 2024/25 Re-Exam

Question 1.2.1: Give an execution diagram for a simple 5-stage pipeline with full forwarding (stages: F,D,X,M,W ... use single letters to make it fit in the boxes) corresponding to the pipeline presented in COD. Assume that backward branches are predicted taken during the D-stage. Explain shortly any additional assumptions you may have to make.

Code			Timing
.L3:	lhu	a5,0(a1)	
	addi	a1,a1,2	
	slli	a5,a5,2	
	add	a5,a2,a5	
	lw	a5,0(a5)	
	add	a0,a0,a5	
	bne	a4,a1,.L3	
.L3:	lhu	a5,0(a1)	
	addi	a1,a1,2	
	slli	a5,a5,2	
	add	a5,a2,a5	
	lw	a5,0(a5)	
	add	a0,a0,a5	
	bne	a4,a1,.L3	

Exam Question 2024/25 Re-Exam

Backward branches are predicted in De stage

Code	Timing																			
.L3																				
lhu a5,0(a1)	F	D	X	M	W															
addi a1,a1,2																				
slli a5, a5, 2																				
add a5, a2, a5																				
lw a5, 0(a5)																				
add a0, a0, a5																				
bne a4, a1, .L3																				
.L3																				
lhu a5,0(a1)																				
addi a1,a1,2																				
slli a5, a5, 2																				
add a5, a2, a5																				
lw a5, 0(a5)																				
add a0, a0, a5																				
bne a4, a1, .L3																				

Exam Question 2024/25 Re-Exam

Backward branches are predicted in De stage

Code	Timing																			
.L3																				
lhu a5,0(a1)	F	D	X	M	W															
addi a1,a1,2		F	D	X	M	W														
slli a5, a5, 2																				
add a5, a2, a5																				
lw a5, 0(a5)																				
add a0, a0, a5																				
bne a4, a1, .L3																				
.L3																				
lhu a5,0(a1)																				
addi a1,a1,2																				
slli a5, a5, 2																				
add a5, a2, a5																				
lw a5, 0(a5)																				
add a0, a0, a5																				
bne a4, a1, .L3																				

Exam Question 2024/25 Re-Exam

Backward branches are predicted in De stage

Code	Timing																			
.L3																				
lhu a5,0(a1)	F	D	X	M	W															
addi a1,a1,2		F	D	X	M	W														
slli a5, a5, 2			F	D	X	M	W													
add a5, a2, a5																				
lw a5, 0(a5)																				
add a0, a0, a5																				
bne a4, a1, .L3																				
.L3																				
lhu a5,0(a1)																				
addi a1,a1,2																				
slli a5, a5, 2																				
add a5, a2, a5																				
lw a5, 0(a5)																				
add a0, a0, a5																				
bne a4, a1, .L3																				

Exam Question 2024/25 Re-Exam

Backward branches are predicted in De stage

Code	Timing																			
.L3																				
lhu a5,0(a1)	F	D	X	M	W															
addi a1,a1,2		F	D	X	M	W														
slli a5, a5, 2			F	D	X	M	W													
add a5, a2, a5				F	D	X	M	W												
lw a5, 0(a5)																				
add a0, a0, a5																				
bne a4, a1, .L3																				
.L3																				
lhu a5,0(a1)																				
addi a1,a1,2																				
slli a5, a5, 2																				
add a5, a2, a5																				
lw a5, 0(a5)																				
add a0, a0, a5																				
bne a4, a1, .L3																				

The result from slli gets forwarded

Exam Question 2024/25 Re-Exam

Backward branches are predicted in De stage

Code	Timing																			
.L3																				
lhu a5,0(a1)	F	D	X	M	W															
addi a1,a1,2		F	D	X	M	W														
slli a5, a5, 2			F	D	X	M	W													
add a5, a2, a5				F	D	X	M	W												
lw a5, 0(a5)					F	D	X	M	W											
add a0, a0, a5																				
bne a4, a1, .L3																				
.L3																				
lhu a5,0(a1)																				
addi a1,a1,2																				
slli a5, a5, 2																				
add a5, a2, a5																				
lw a5, 0(a5)																				
add a0, a0, a5																				
bne a4, a1, .L3																				

Exam Question 2024/25 Re-Exam

Backward branches are predicted in De stage

Code	Timing																			
.L3																				
lhu a5,0(a1)	F	D	X	M	W															
addi a1,a1,2		F	D	X	M	W														
slli a5, a5, 2			F	D	X	M	W													
add a5, a2, a5				F	D	X	M	W												
lw a5, 0(a5)					F	D	X	M	W											
add a0, a0, a5						F	D	D	X	M	W									
bne a4, a1, .L3																				
.L3																				
lhu a5,0(a1)																				
addi a1,a1,2																				
slli a5, a5, 2																				
add a5, a2, a5																				
lw a5, 0(a5)																				
add a0, a0, a5																				
bne a4, a1, .L3																				

Stall until a4 is forwarded from lw's M stage

Exam Question 2024/25 Re-Exam

Backward branches are predicted in De stage

Code	Timing																			
.L3																				
lhu a5,0(a1)	F	D	X	M	W															
addi a1,a1,2		F	D	X	M	W														
slli a5, a5, 2			F	D	X	M	W													
add a5, a2, a5				F	D	X	M	W												
lw a5, 0(a5)					F	D	X	M	W											
add a0, a0, a5						F	D	D	X	M	W									
bne a4, a1, .L3							F	F	D	X	M	W								
.L3																				
lhu a5,0(a1)																				
addi a1,a1,2																				
slli a5, a5, 2																				
add a5, a2, a5																				
lw a5, 0(a5)																				
add a0, a0, a5																				
bne a4, a1, .L3																				

Exam Question 2024/25 Re-Exam

Backward branches are predicted in De stage

Code	Timing																			
.L3																				
lhu a5,0(a1)	F	D	X	M	W															
addi a1,a1,2		F	D	X	M	W														
slli a5, a5, 2			F	D	X	M	W													
add a5, a2, a5				F	D	X	M	W												
lw a5, 0(a5)					F	D	X	M	W											
add a0, a0, a5						F	D	D	X	M	W									
bne a4, a1, .L3							F	F	D	X	M	W								
.L3																				
lhu a5,0(a1)										F	D	X	M	W						
addi a1,a1,2																				
slli a5, a5, 2																				
add a5, a2, a5																				
lw a5, 0(a5)																				
add a0, a0, a5																				
bne a4, a1, .L3																				

Correctly predicted backward branch

Exam Question 2024/25 Re-Exam

Backward branches are predicted in De stage

Code	Timing																			
.L3																				
lhu a5,0(a1)	F	D	X	M	W															
addi a1,a1,2		F	D	X	M	W														
slli a5, a5, 2			F	D	X	M	W													
add a5, a2, a5				F	D	X	M	W												
lw a5, 0(a5)					F	D	X	M	W											
add a0, a0, a5						F	D	D	X	M	W									
bne a4, a1, .L3							F	F	D	X	M	W								
.L3																				
lhu a5,0(a1)										F	D	X	M	W						
addi a1,a1,2											F	D	X	M	W					
slli a5, a5, 2												F	D	X	M	W				
add a5, a2, a5													F	D	X	M	W			
lw a5, 0(a5)														F	D	X	M	W		
add a0, a0, a5															F	D	D	X	M	W
bne a4, a1, .L3																F	F	D	X	M	W

Exam Question 2024/25 Re-Exam

Question 1.2.2:

Consider the following incomplete execution diagram for an out-of-order machine with the following resources:

- Pipeline frontend (Fa to Qu) and commit (Ca,Cb) can handle 4 instructions/clock
- Predictor can predict 1 branch/clock
- Out-of-order section can handle 1 arithmetic operation/clock
- Out-of-order section can handle 1 load or store operation/clock
- Out-of-order section can handle 1 control-flow operation/clock

```
1  .L3:    lhu     a5,0(a1)    Fa Fb Fc De Fu Al Rn Qu pk rd ag ma mb mc wr Ca Cb
2          addi    a1,a1,2     Fa Fb Fc De Fu Al Rn Qu pk rd ex wr          Ca Cb
3          slli    a5,a5,2     Fa Fb Fc De Fu Al Rn Qu -- -- -- -- pk rd ex wr Ca Cb
4          add     a5,a2,a5    Fa Fb Fc De Fu Al Rn Qu -- -- -- -- -- pk rd ex wr Ca Cb
5          lw      a5,0(a5)       Fa Fb Fc De Fu Al Rn Qu -- -- -- -- -- pk rd ag ma mb mc wr Ca Cb
6          add     a0,a0,a5       Fa Fb Fc De Fu Al Rn Qu -- -- -- -- -- -- -- -- -- pk rd ex wr Ca Cb
7          bne     a4,a1,.L3      Fa Fb Fc De Fu Al Rn Qu pk rd ex wr                              Ca Cb
8  .L3:    lhu     a5,0(a1)          Fa Fb Fc De Fu Al Rn Qu pk rd ag ma mb mc wr                  Ca Cb
9          addi    a1,a1,2           Fa Fb Fc De Fu Al Rn Qu pk rd ex wr                           Ca Cb
10         slli    a5,a5,2           Fa Fb Fc De Fu Al Rn Qu -- -- -- -- pk rd ex wr                  Ca Cb
11         add     a5,a2,a5          Fa Fb Fc De Fu Al Rn Qu -- -- -- -- -- pk rd ex wr               Ca Cb
12         lw      a5,0(a5)             Fa Fb Fc De Fu Al Rn Qu ?? ?? ?? ?? ?? ?? ?? ?? ?? ?? ?? ?? ??
13         add     a0,a0,a5             Fa Fb Fc De Fu Al Rn Qu ?? ?? ?? ?? ?? ?? ?? ?? ?? ?? ?? ?? ?? ??
14         bne     a4,a1,.L3            Fa Fb Fc De Fu Al Rn Qu ?? ?? ?? ?? ?? ?? ?? ?? ?? ?? ?? ?? ?? ??
```

Complete the execution diagram by presenting the section marked with "??" here:
Explain shortly any additional assumptions you may have to make.

Exam Question 2024/25 Re-Exam

- Pipeline frontend (Fa to Qu) and commit (Ca,Cb) can handle 4 instructions/clock
- Predictor can predict 1 branch/clock
- Out-of-order section can handle 1 arithmetic operation/clock
- Out-of-order section can handle 1 load or store operation/clock
- Out-of-order section can handle 1 control-flow operation/clock

```
1  .L3:    lhu     a5,0(a1)    Fa Fb Fc De Fu Al Rn Qu pk rd ag ma mb mc wr Ca Cb
2          addi    a1,a1,2     Fa Fb Fc De Fu Al Rn Qu pk rd ex wr          Ca Cb
3          slli    a5,a5,2     Fa Fb Fc De Fu Al Rn Qu -- -- -- -- pk rd ex wr Ca Cb
4          add     a5,a2,a5    Fa Fb Fc De Fu Al Rn Qu -- -- -- -- -- pk rd ex wr Ca Cb
5          lw      a5,0(a5)       Fa Fb Fc De Fu Al Rn Qu -- -- -- -- -- pk rd ag ma mb mc wr Ca Cb
6          add     a0,a0,a5       Fa Fb Fc De Fu Al Rn Qu -- -- -- -- -- -- -- -- -- pk rd ex wr Ca Cb
7          bne     a4,a1,.L3      Fa Fb Fc De Fu Al Rn Qu pk rd ex wr                           Ca Cb
8  .L3:    lhu     a5,0(a1)          Fa Fb Fc De Fu Al Rn Qu pk rd ag ma mb mc wr                Ca Cb
9          addi    a1,a1,2           Fa Fb Fc De Fu Al Rn Qu pk rd ex wr                        Ca Cb
10         slli    a5,a5,2           Fa Fb Fc De Fu Al Rn Qu -- -- -- -- pk rd ex wr               Ca Cb
11         add     a5,a2,a5          Fa Fb Fc De Fu Al Rn Qu -- -- -- -- -- pk rd ex wr            Ca Cb
12         lw      a5,0(a5)             Fa Fb Fc De Fu Al Rn Qu ?? ?? ?? ?? ?? ?? ?? ?? ?? ?? ?? ?? ?? ??
13         add     a0,a0,a5             Fa Fb Fc De Fu Al Rn Qu ?? ?? ?? ?? ?? ?? ?? ?? ?? ?? ?? ?? ?? ?? ??
14         bne     a4,a1,.L3            Fa Fb Fc De Fu Al Rn Qu ?? ?? ?? ?? ?? ?? ?? ?? ?? ?? ?? ?? ?? ?? ??
```

ALU-op: 1 clock cycle latency
Load-op: 4 clock cycles latency

Code	Timing															
lw a5, 0(a5)	Qu															
add a0, a0, a5	Qu															
bne a4, a1, .L3	Qu															

Exam Question 2024/25 Re-Exam

- Pipeline frontend (Fa to Qu) and commit (Ca,Cb) can handle 4 instructions/clock
- Predictor can predict 1 branch/clock
- Out-of-order section can handle 1 arithmetic operation/clock
- Out-of-order section can handle 1 load or store operation/clock
- Out-of-order section can handle 1 control-flow operation/clock

```
1  .L3:   lhu    a5,0(a1)    Fa Fb Fc De Fu Al Rn Qu pk rd ag ma mb mc wr Ca Cb
2         addi   a1,a1,2     Fa Fb Fc De Fu Al Rn Qu pk rd ex wr          Ca Cb
3         slli   a5,a5,2     Fa Fb Fc De Fu Al Rn Qu -- -- -- -- pk rd ex wr Ca Cb
4         add    a5,a2,a5    Fa Fb Fc De Fu Al Rn Qu -- -- -- -- -- pk rd ex wr Ca Cb
5         lw     a5,0(a5)       Fa Fb Fc De Fu Al Rn Qu -- -- -- -- -- pk rd ag ma mb mc wr Ca Cb
6         add    a0,a0,a5       Fa Fb Fc De Fu Al Rn Qu -- -- -- -- -- -- -- -- -- pk rd ex wr Ca Cb
7         bne    a4,a1,.L3      Fa Fb Fc De Fu Al Rn Qu pk rd ex wr                            Ca Cb
8  .L3:   lhu    a5,0(a1)          Fa Fb Fc De Fu Al Rn Qu pk rd ag ma mb mc wr                Ca Cb
9         addi   a1,a1,2           Fa Fb Fc De Fu Al Rn Qu pk rd ex wr                         Ca Cb
10        slli   a5,a5,2           Fa Fb Fc De Fu Al Rn Qu -- -- -- -- pk rd ex wr                Ca Cb
11        add    a5,a2,a5          Fa Fb Fc De Fu Al Rn Qu -- -- -- -- -- pk rd ex wr             Ca Cb
12        lw     a5,0(a5)             Fa Fb Fc De Fu Al Rn Qu ?? ?? ?? ?? ?? ?? ?? ?? ?? ?? ?? ?? ?? ??
13        add    a0,a0,a5             Fa Fb Fc De Fu Al Rn Qu ?? ?? ?? ?? ?? ?? ?? ?? ?? ?? ?? ?? ?? ?? ??
14        bne    a4,a1,.L3            Fa Fb Fc De Fu Al Rn Qu ?? ?? ?? ?? ?? ?? ?? ?? ?? ?? ?? ?? ?? ?? ??
```

ALU-op: 1 clock cycle latency
Load-op: 4 clock cycles latency

Code	Timing															
lw a5, 0(a5)	Qu															
add a0, a0, a5	Qu															
bne a4, a1, .L3	Qu															

Exam Question 2024/25 Re-Exam

- Pipeline frontend (Fa to Qu) and commit (Ca,Cb) can handle 4 instructions/clock
- Predictor can predict 1 branch/clock
- Out-of-order section can handle 1 arithmetic operation/clock
- Out-of-order section can handle 1 load or store operation/clock
- Out-of-order section can handle 1 control-flow operation/clock

```
1  .L3:   lhu    a5,0(a1)    Fa Fb Fc De Fu Al Rn Qu pk rd ag ma mb mc wr Ca Cb
2         addi   a1,a1,2     Fa Fb Fc De Fu Al Rn Qu pk rd ex wr          Ca Cb
3         slli   a5,a5,2     Fa Fb Fc De Fu Al Rn Qu -- -- -- -- pk rd ex wr Ca Cb
4         add    a5,a2,a5    Fa Fb Fc De Fu Al Rn Qu -- -- -- -- -- pk rd ex wr Ca Cb
5         lw     a5,0(a5)       Fa Fb Fc De Fu Al Rn Qu -- -- -- -- -- pk rd ag ma mb mc wr Ca Cb
6         add    a0,a0,a5       Fa Fb Fc De Fu Al Rn Qu -- -- -- -- -- -- -- -- -- pk rd ex wr Ca Cb
7         bne    a4,a1,.L3      Fa Fb Fc De Fu Al Rn Qu pk rd ex wr                            Ca Cb
8  .L3:   lhu    a5,0(a1)          Fa Fb Fc De Fu Al Rn Qu pk rd ag ma mb mc wr                Ca Cb
9         addi   a1,a1,2           Fa Fb Fc De Fu Al Rn Qu pk rd ex wr                         Ca Cb
10        slli   a5,a5,2           Fa Fb Fc De Fu Al Rn Qu -- -- -- -- pk rd ex wr                Ca Cb
11        add    a5,a2,a5          Fa Fb Fc De Fu Al Rn Qu -- -- -- -- -- pk rd ex wr             Ca Cb
12        lw     a5,0(a5)             Fa Fb Fc De Fu Al Rn Qu ?? ?? ?? ?? ?? ?? ?? ?? ?? ?? ?? ?? ?? ??
13        add    a0,a0,a5             Fa Fb Fc De Fu Al Rn Qu ?? ?? ?? ?? ?? ?? ?? ?? ?? ?? ?? ?? ?? ?? ??
14        bne    a4,a1,.L3            Fa Fb Fc De Fu Al Rn Qu ?? ?? ?? ?? ?? ?? ?? ?? ?? ?? ?? ?? ?? ?? ??
```

ALU-op: 1 clock cycle latency
Load-op: 4 clock cycles latency

Code	Timing															
lw a5, 0(a5)	Qu	--	--	--	--	--	pk	rd	ag	ma	mb	mc	wr	Ca	Cb	
add a0, a0, a5	Qu															
bne a4, a1, .L3	Qu															

Exam Question 2024/25 Re-Exam

- Pipeline frontend (Fa to Qu) and commit (Ca,Cb) can handle 4 instructions/clock
- Predictor can predict 1 branch/clock
- Out-of-order section can handle 1 arithmetic operation/clock
- Out-of-order section can handle 1 load or store operation/clock
- Out-of-order section can handle 1 control-flow operation/clock

```
1  .L3:   lhu    a5,0(a1)    Fa Fb Fc De Fu Al Rn Qu pk rd ag ma mb mc wr Ca Cb
2         addi   a1,a1,2     Fa Fb Fc De Fu Al Rn Qu pk rd ex wr          Ca Cb
3         slli   a5,a5,2     Fa Fb Fc De Fu Al Rn Qu -- -- -- -- pk rd ex wr Ca Cb
4         add    a5,a2,a5    Fa Fb Fc De Fu Al Rn Qu -- -- -- -- -- pk rd ex wr Ca Cb
5         lw     a5,0(a5)       Fa Fb Fc De Fu Al Rn Qu -- -- -- -- -- pk rd ag ma mb mc wr Ca Cb
6         add    a0,a0,a5       Fa Fb Fc De Fu Al Rn Qu -- -- -- -- -- -- -- -- -- pk rd ex wr Ca Cb
7         bne    a4,a1,.L3      Fa Fb Fc De Fu Al Rn Qu pk rd ex wr                        Ca Cb
8  .L3:   lhu    a5,0(a1)          Fa Fb Fc De Fu Al Rn Qu pk rd ag ma mb mc wr             Ca Cb
9         addi   a1,a1,2           Fa Fb Fc De Fu Al Rn Qu pk rd ex wr                     Ca Cb
10        slli   a5,a5,2           Fa Fb Fc De Fu Al Rn Qu -- -- -- -- pk rd ex wr            Ca Cb
11        add    a5,a2,a5          Fa Fb Fc De Fu Al Rn Qu -- -- -- -- -- pk rd ex wr         Ca Cb
12        lw     a5,0(a5)             Fa Fb Fc De Fu Al Rn Qu ?? ?? ?? ?? ?? ?? ?? ?? ?? ?? ?? ?? ?? ?? ??
13        add    a0,a0,a5             Fa Fb Fc De Fu Al Rn Qu ?? ?? ?? ?? ?? ?? ?? ?? ?? ?? ?? ?? ?? ?? ??
14        bne    a4,a1,.L3            Fa Fb Fc De Fu Al Rn Qu ?? ?? ?? ?? ?? ?? ?? ?? ?? ?? ?? ?? ?? ?? ??
```

ALU-op: 1 clock cycle latency
Load-op: 4 clock cycles latency

Code	Timing															
lw a5, 0(a5)	Qu	--	--	--	--	--	pk	rd	ag	ma	mb	mc	wr	Ca	Cb	
add a0, a0, a5	Qu															
bne a4, a1, .L3	Qu															

Exam Question 2024/25 Re-Exam

- Pipeline frontend (Fa to Qu) and commit (Ca,Cb) can handle 4 instructions/clock
- Predictor can predict 1 branch/clock
- Out-of-order section can handle 1 arithmetic operation/clock
- Out-of-order section can handle 1 load or store operation/clock
- Out-of-order section can handle 1 control-flow operation/clock

```
1  .L3:   lhu    a5,0(a1)    Fa Fb Fc De Fu Al Rn Qu pk rd ag ma mb mc wr Ca Cb
2         addi   a1,a1,2     Fa Fb Fc De Fu Al Rn Qu pk rd ex wr          Ca Cb
3         slli   a5,a5,2     Fa Fb Fc De Fu Al Rn Qu -- -- -- -- pk rd ex wr Ca Cb
4         add    a5,a2,a5    Fa Fb Fc De Fu Al Rn Qu -- -- -- -- -- pk rd ex wr Ca Cb
5         lw     a5,0(a5)       Fa Fb Fc De Fu Al Rn Qu -- -- -- -- -- pk rd ag ma mb mc wr Ca Cb
6         add    a0,a0,a5       Fa Fb Fc De Fu Al Rn Qu -- -- -- -- -- -- -- -- -- pk rd ex wr Ca Cb
7         bne    a4,a1,.L3      Fa Fb Fc De Fu Al Rn Qu pk rd ex wr                        Ca Cb
8  .L3:   lhu    a5,0(a1)          Fa Fb Fc De Fu Al Rn Qu pk rd ag ma mb mc wr             Ca Cb
9         addi   a1,a1,2           Fa Fb Fc De Fu Al Rn Qu pk rd ex wr                      Ca Cb
10        slli   a5,a5,2           Fa Fb Fc De Fu Al Rn Qu -- -- -- -- pk rd ex wr             Ca Cb
11        add    a5,a2,a5          Fa Fb Fc De Fu Al Rn Qu -- -- -- -- -- pk rd ex wr          Ca Cb
12        lw     a5,0(a5)             Fa Fb Fc De Fu Al Rn Qu ?? ?? ?? ?? ?? ?? ?? ?? ?? ?? ?? ?? ?? ??
13        add    a0,a0,a5             Fa Fb Fc De Fu Al Rn Qu ?? ?? ?? ?? ?? ?? ?? ?? ?? ?? ?? ?? ?? ?? ??
14        bne    a4,a1,.L3            Fa Fb Fc De Fu Al Rn Qu ?? ?? ?? ?? ?? ?? ?? ?? ?? ?? ?? ?? ?? ?? ??
```

ALU-op: 1 clock cycle latency
Load-op: 4 clock cycles latency

Code	Timing															
lw a5, 0(a5)	Qu	--	--	--	--	--	pk	rd	ag	ma	mb	mc	wr	Ca	Cb	
add a0, a0, a5	Qu	--	--	--	--	--	--	--	--	--	pk	rd	ex	wr	Ca	Cb
bne a4, a1, .L3	Qu															

Exam Question 2024/25 Re-Exam

- Pipeline frontend (Fa to Qu) and commit (Ca,Cb) can handle 4 instructions/clock
- Predictor can predict 1 branch/clock
- Out-of-order section can handle 1 arithmetic operation/clock
- Out-of-order section can handle 1 load or store operation/clock
- Out-of-order section can handle 1 control-flow operation/clock

```
1  .L3:   lhu    a5,0(a1)    Fa Fb Fc De Fu Al Rn Qu pk rd ag ma mb mc wr Ca Cb
2         addi   a1,a1,2     Fa Fb Fc De Fu Al Rn Qu pk rd ex wr          Ca Cb
3         slli   a5,a5,2     Fa Fb Fc De Fu Al Rn Qu -- -- -- -- pk rd ex wr Ca Cb
4         add    a5,a2,a5    Fa Fb Fc De Fu Al Rn Qu -- -- -- -- -- pk rd ex wr Ca Cb
5         lw     a5,0(a5)       Fa Fb Fc De Fu Al Rn Qu -- -- -- -- -- pk rd ag ma mb mc wr Ca Cb
6         add    a0,a0,a5       Fa Fb Fc De Fu Al Rn Qu -- -- -- -- -- -- -- -- -- pk rd ex wr Ca Cb
7         bne    a4,a1,.L3      Fa Fb Fc De Fu Al Rn Qu pk rd ex wr                           Ca Cb
8  .L3:   lhu    a5,0(a1)          Fa Fb Fc De Fu Al Rn Qu pk rd ag ma mb mc wr                Ca Cb
9         addi   a1,a1,2           Fa Fb Fc De Fu Al Rn Qu pk rd ex wr                        Ca Cb
10        slli   a5,a5,2           Fa Fb Fc De Fu Al Rn Qu -- -- -- -- pk rd ex wr               Ca Cb
11        add    a5,a2,a5          Fa Fb Fc De Fu Al Rn Qu -- -- -- -- -- pk rd ex wr            Ca Cb
12        lw     a5,0(a5)             Fa Fb Fc De Fu Al Rn Qu ?? ?? ?? ?? ?? ?? ?? ?? ?? ?? ?? ?? ?? ??
13        add    a0,a0,a5             Fa Fb Fc De Fu Al Rn Qu ?? ?? ?? ?? ?? ?? ?? ?? ?? ?? ?? ?? ?? ?? ??
14        bne    a4,a1,.L3            Fa Fb Fc De Fu Al Rn Qu ?? ?? ?? ?? ?? ?? ?? ?? ?? ?? ?? ?? ?? ?? ??
```

ALU-op: 1 clock cycle latency
Load-op: 4 clock cycles latency

Code	Timing															
lw a5, 0(a5)	Qu	--	--	--	--	--	pk	rd	ag	ma	mb	mc	wr	Ca	Cb	
add a0, a0, a5	Qu	--	--	--	--	--	--	--	--	--	pk	rd	ex	wr	Ca	Cb
bne a4, a1, .L3	Qu															

Exam Question 2024/25 Re-Exam

- Pipeline frontend (Fa to Qu) and commit (Ca,Cb) can handle 4 instructions/clock
- Predictor can predict 1 branch/clock
- Out-of-order section can handle 1 arithmetic operation/clock
- Out-of-order section can handle 1 load or store operation/clock
- Out-of-order section can handle 1 control-flow operation/clock

```
1  .L3:   lhu    a5,0(a1)    Fa Fb Fc De Fu Al Rn Qu pk rd ag ma mb mc wr Ca Cb
2         addi   a1,a1,2     Fa Fb Fc De Fu Al Rn Qu pk rd ex wr          Ca Cb
3         slli   a5,a5,2     Fa Fb Fc De Fu Al Rn Qu -- -- -- -- pk rd ex wr Ca Cb
4         add    a5,a2,a5    Fa Fb Fc De Fu Al Rn Qu -- -- -- -- -- pk rd ex wr Ca Cb
5         lw     a5,0(a5)       Fa Fb Fc De Fu Al Rn Qu -- -- -- -- -- pk rd ag ma mb mc wr Ca Cb
6         add    a0,a0,a5       Fa Fb Fc De Fu Al Rn Qu -- -- -- -- -- -- -- -- -- pk rd ex wr Ca Cb
7         bne    a4,a1,.L3      Fa Fb Fc De Fu Al Rn Qu pk rd ex wr                           Ca Cb
8  .L3:   lhu    a5,0(a1)          Fa Fb Fc De Fu Al Rn Qu pk rd ag ma mb mc wr                Ca Cb
9         addi   a1,a1,2           Fa Fb Fc De Fu Al Rn Qu pk rd ex wr                        Ca Cb
10        slli   a5,a5,2           Fa Fb Fc De Fu Al Rn Qu -- -- -- -- pk rd ex wr               Ca Cb
11        add    a5,a2,a5          Fa Fb Fc De Fu Al Rn Qu -- -- -- -- -- pk rd ex wr            Ca Cb
12        lw     a5,0(a5)             Fa Fb Fc De Fu Al Rn Qu ?? ?? ?? ?? ?? ?? ?? ?? ?? ?? ?? ?? ?? ??
13        add    a0,a0,a5             Fa Fb Fc De Fu Al Rn Qu ?? ?? ?? ?? ?? ?? ?? ?? ?? ?? ?? ?? ?? ?? ??
14        bne    a4,a1,.L3            Fa Fb Fc De Fu Al Rn Qu ?? ?? ?? ?? ?? ?? ?? ?? ?? ?? ?? ?? ?? ?? ??
```

ALU-op: 1 clock cycle latency
Load-op: 4 clock cycles latency

Code	Timing															
lw a5, 0(a5)	Qu	--	--	--	--	--	pk	rd	ag	ma	mb	mc	wr	Ca	Cb	
add a0, a0, a5	Qu	--	--	--	--	--	--	--	--	--	pk	rd	ex	wr	Ca	Cb
bne a4, a1, .L3	Qu															

Exam Question 2024/25 Re-Exam

- Pipeline frontend (Fa to Qu) and commit (Ca,Cb) can handle 4 instructions/clock
- Predictor can predict 1 branch/clock
- Out-of-order section can handle 1 arithmetic operation/clock
- Out-of-order section can handle 1 load or store operation/clock
- Out-of-order section can handle 1 control-flow operation/clock

```
1  .L3:   lhu    a5,0(a1)    Fa Fb Fc De Fu Al Rn Qu pk rd ag ma mb mc wr Ca Cb
2         addi   a1,a1,2     Fa Fb Fc De Fu Al Rn Qu pk rd ex wr          Ca Cb
3         slli   a5,a5,2     Fa Fb Fc De Fu Al Rn Qu -- -- -- -- pk rd ex wr Ca Cb
4         add    a5,a2,a5    Fa Fb Fc De Fu Al Rn Qu -- -- -- -- -- pk rd ex wr Ca Cb
5         lw     a5,0(a5)       Fa Fb Fc De Fu Al Rn Qu -- -- -- -- -- pk rd ag ma mb mc wr Ca Cb
6         add    a0,a0,a5       Fa Fb Fc De Fu Al Rn Qu -- -- -- -- -- -- -- -- -- pk rd ex wr Ca Cb
7         bne    a4,a1,.L3      Fa Fb Fc De Fu Al Rn Qu pk rd ex wr                           Ca Cb
8  .L3:   lhu    a5,0(a1)          Fa Fb Fc De Fu Al Rn Qu pk rd ag ma mb mc wr                Ca Cb
9         addi   a1,a1,2           Fa Fb Fc De Fu Al Rn Qu pk rd ex wr                        Ca Cb
10        slli   a5,a5,2           Fa Fb Fc De Fu Al Rn Qu -- -- -- -- pk rd ex wr               Ca Cb
11        add    a5,a2,a5          Fa Fb Fc De Fu Al Rn Qu -- -- -- -- -- pk rd ex wr            Ca Cb
12        lw     a5,0(a5)             Fa Fb Fc De Fu Al Rn Qu ?? ?? ?? ?? ?? ?? ?? ?? ?? ?? ?? ?? ?? ??
13        add    a0,a0,a5             Fa Fb Fc De Fu Al Rn Qu ?? ?? ?? ?? ?? ?? ?? ?? ?? ?? ?? ?? ?? ?? ??
14        bne    a4,a1,.L3            Fa Fb Fc De Fu Al Rn Qu ?? ?? ?? ?? ?? ?? ?? ?? ?? ?? ?? ?? ?? ?? ??
```

ALU-op: 1 clock cycle latency
Load-op: 4 clock cycles latency

Code	Timing															
lw a5, 0(a5)	Qu	--	--	--	--	--	pk	rd	ag	ma	mb	mc	wr	Ca	Cb	
add a0, a0, a5	Qu	--	--	--	--	--	--	--	--	--	pk	rd	ex	wr	Ca	Cb
bne a4, a1, .L3	Qu	pk	rd	ex	wr	--	--	--	--	--	--	--	--	--	Ca	Cb


THE GREAT GATSBY
DIKU GALA
2026