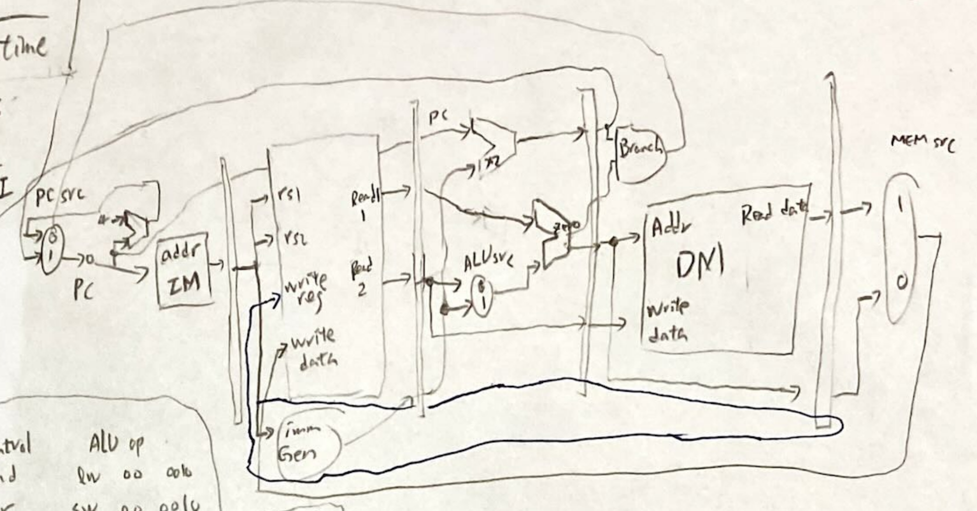
$CPU\ time = \frac{No.instr \times CPI}{clock\ rate(f)}$

$clock\ rate(f) = \frac{1}{cycle\ time}$

IF　ID　EX　MEM　WB



$No.instr = \frac{clock\ rate \times time}{CPI}$

$clock\ cycle = No.A \cdot CPI_A + No.B \cdot CPI_B$

$CPI(total) = \frac{clock\ cycle}{No.total}$

1's versus of 2's $= \frac{f_2}{f_1}$

speed up A to B $\frac{V_A}{V_B} = \frac{T_B}{T_A}$

MIPS $= \frac{f}{10^6 \times CPI}$

SPEC ratio $= \frac{ref.time}{exe.time}$

multi-cores

arith　l/s　branch

1　$\frac{No \times CPI + No \times CPI + No \times CPI}{f}$

2　$\frac{\frac{No \times CPI}{2 \times 0.7} + \frac{No \times CPI}{2 \times 0.7} + No \times CPI}{f}$

+4s　$Time = \frac{t}{P}$

	Time	real-time	speedup	actual speedup
1	100			
2	50	54	$\frac{100}{54} = 1.85$	$\frac{1.85}{2} = 0.93$
4	25	29	$\frac{100}{29} = 3.44$	$\frac{3.44}{4} = 0.86$

ALU control
- 0000 and
- 0001 or
- 0010 +
- 0110 −

ALU op
- lw 00 0010
- sw 00 0010
- beq 01 0110
- R 10

消耗能量 $\frac{1}{ps} \times \% \times J$

R-type 不用 sign 1st (add, or)

B-type imm 省略 0bit

Branch mux　PC + imm × 2

Predict，要 3 個 flush (IF, ID, EX)

beq % × (1 − accuracy) × 3 + 1

speedup

CPU　Branch % (1 − accuracy) + 1

stuck (ALU src)

at zero (3 個奇數)

add x3, x1, x1

⇒ 跑到 x2，not x3

stuck at 1

add x2, x1, x1

⇒ 跑到 x3，not x2

sol (1) 放到沒用的 reg

(2) 放到 stack

MEM read stuck

ld ..(0) 不一定能一次抓出來

Pipeline cycle (minimum)

no. stage　n + K − 1，看 WB，K cycle 得 1st instr，剩下 n−1 instr

stall 放 ID 後

FP　l/s　branch　INT

0x abcdef 12

	3	2	1	0
litt end	ab	cd	ef	12
Big end	12	ef	cd	ab

0x8 + 0xD = 0x5 (overflow)

0x8 − 0xD = (−5) = 0xB (no overflow)　16 − 5 = 11 = B

jal range [0x1ff0000 0, 0x200FFFFE] $(-2^{20} \sim 2^{20} - 1)$ (相對位置)

Forwarding

EX to 1st

add x11, x12, x13

add x14, x11, x15

MEM 1st

lw x11, 0(x1)

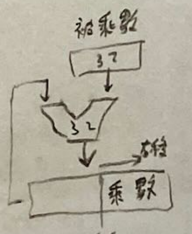
add x15, x11, x13

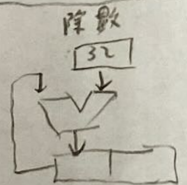
EX 1st & EX 2nd

add x11, x12, x13

add x5, x11, x15

add x16, x11, x12





no forwarding

1st % × 2 + 1st % × 2 + 2nd % × 1 + 2nd % × 1 (EX, MEM, EX, MEM)

+ 1st & 2nd % × 2 = stall (0x0.85)

CPI' = CPI(1) + 0.85

$\frac{0.85}{1.85} = 46\%$ on stall

Bias　$2^7 - 1 = 127$　$2^{10} - 1 = 1023$

S	Exponent (8 / 11)	fraction (23 / 52)

$(-1)^S \times (1 + Fraction) \times 2^{Exponent - Bias}$

precise 2^{-23}　$23 \times \log_{10}2 = 23 \times 0.3 \doteq 6$ decimal

Denormal　$(0 + fraction) \times 2^{1 - Bias}$

±∞ ⇒ exponent 11111　fraction 0000

NaN ⇒ exponent 1111　fraction ≠ 0000

加法　fraction >>

乘法　ex. $2^1 \times 2^2$　exponent −1−2 + 127

乘法器　(bit − 1) × T

multi　$\log_2 bit \times T$

硬　3 cycle × 8 × T

軟　5 × 8 × T

Full forwarding

MEM 1st % × 1　其餘 = 0

Forward EX/MEM　A = 10

EX 1st = 0，MEM 1st = 2

EX 2nd = 1，MEM 2nd = 1

EX 1 & 2 = 1

Forward MEM/WB　A = 01

EX 1st = 1，MEM 1st = 1

EX 2nd = 0，MEM 2nd = 0

EX 1 & 2 = 1

temporal locality 常被用 I, J, B[I][0]

satial : 空間近 A[I][J]

(for 迴圈內)

B[I][0] I=1~8 ng n1000 A[I][J]

$8 \times 1000 \times \frac{64\ bit}{16\ (byte)} = 4000$ (main)

Matlab $8 \times \frac{64\ bit}{16\ (byte)} = 4$ (block)

C 8 (block)

write through 寫回下一層 (bffer, mem)

non allocate 跳過 buffer, ~~mem~~ (拉 non-資料回來)

write back 寫 buffer (看 dirty)

write allocate mem → 寫 buffer → 寫 catch

| tag | index | (off set) 根據 n-way

ex. 16 one-word block

16 block (index = 4 bit)

n-word (set), off-set = $\log_2 n$ bit

先看 index

tag + index Φ = hit

addr (Tag錯 index對) Miss

0x00	0x0	0x0B	hit (不管 off set)
0x10	0x0	0x00	miss
0x400	0x1	0x00	⇒ 0x00 ⊕ 0x1F + 整個 index 都是1 (5bit = 0x1F)
0x1e	0x00	0x00	miss ⇒ 0x400 ⊕ 0x1F

Optimiz (8 bit data) 5 tag 3 (index + off set)

total cycle = No. miss × T + No. total [nword+1]

clock rate = $\frac{1}{hit\ time}$

CPI: $\frac{main\ access\ time}{Hit\ time}$ = cycles (整數)

T: 1 + miss rate × cycles (cycle)

CPI total

1 + miss rate × cycle + % access / inst × miss rate × cycle

Time per inst = CPI total × hit time

$\frac{L_2\ hit\ time}{L_1\ hit\ time}$ = cycle

$1 + L_1$ miss rate [+ L_2 miss rate ×]

= AMAT

Way 0 Way 1

T(index) = (tag) T(index) = (tag')

ex. 32 bit 2-word block (32 addr - 32 data)

$8\ (byte) \times 2 = 2^4$ (byte)

32 KiB = 2^{15}

lines = $\frac{2^{15}}{2^4} = 2^{11}$ (lines)

(8 byte data off) L2-word = 1 (off) (16 word = 4 off)

"49" = 64 (bit) − 3 − 1 − 11

sizes $2^{15} \times 8 + 2^{11} \times 49 + 2^{11} \times 1$

(data) (tag) (valid)

ex. (32 addr - 32 data)

Tag	index	off
63~10	9-5	4~0

5-3 (8 byte 個bit數)

store: $2^5 \times 2 \times 64$ (bits) = □

(block line) (block size in word)

total = □ + 54×2^5 + 1×2^5 (bits)

(tag) (valid)

optimal size for miss latency ex. 20×B

8: % (20×8) 取小的 AMAT

16: % (20×16)

$\frac{1}{CPI}$ = instr. per cycle

instr. × data r/w % × miss rate % × Block size

instr miss

{ read → data cache miss

write → data cache miss

(write traffic: $\frac{1}{CPI}$ × write × 8 (1 word)

5.12.1 [10] <§5.4> Calculate the CPI for the processor in the table using: 1) only a first-level cache, 2) a second-level direct-mapped cache, and 3) a second-level eight-way set associative cache. How do these numbers change if main memory access time doubles? (Give each change as both an absolute CPI and a percent change.) Notice the extent to which an L2 cache can hide the effects of a slow memory.

CPI

5.12.1 Standard memory time: Each cycle on a 2-Ghz machine takes 0.5 ps. Thus a main memory access requires 100/0.5 = 200 cycles.

- L1 only: 1.5 + 0.07*200 = 15.5
- Direct mapped L2: 1.5 + .07 × (12 + 0.035 × 200) = 2.83
- 8-way set associated L2: 1.5 + .07 × (28 + 0.015 × 200) = 3.67.

Doubled memory access time (thus, a main memory access requires 400 cycles)

- L1 only: 1.5 + 0.07*400 = 29.5 (90% increase)
- Direct mapped L2: 1.5 + .07 × (12 + 0.035 × 400) = 3.32 (17% increase)
- 8-way set associated L2: 1.5 + .07 × (28 + 0.015 × 400) = 3.88 (5% increase)

5.12.2 1.5 = 0.07 × (12 + 0.035 × (50 + 0.13 × 200)) = 1.03

TLB

Valid	Tag	Physical Page Number	Time Since Last Access
1	0xb	12	4
1	0x7	4	1
1	0x3	6	3
0	0x4	9	7

Page table

Index	Valid	Physical Page or in Disk
0	1	5
1	0	Disk
2	0	Disk
3	1	6
4	1	9
5	1	11
6	0	Disk
7	1	4
8	0	Disk
9	0	Disk
a	1	3
b	1	12

5.16

5.16.1

Address	Virtual Page	TLB H/M	TLB		
			Valid	Tag	Physical Page
4669 0x123d	1	TLB miss PT hit PF	1	b	12
			1	7	4
			1	3	6
			1 (last access 0)	1	13
2227 0x08b3	0	TLB miss PT hit	1 (last access 1)	0	5
			1	7	4
			1	3	6
			1 (last access 0)	1	13
13916 0x365c	3	TLB miss PT hit	1 (last access 1)	0	5
			1	7	4
			1 (last access 2)	3	6
			1 (last access 0)	1	13
34587 0x871b	8	TLB miss PT hit PF	1 (last access 1)	0	5
			1 (last access 3)	8	14
			1 (last access 2)	3	6
			1 (last access 0)	1	13
48870 0xbee6	b	TLB miss PT hit	1 (last access 1)	0	5
			1 (last access 3)	8	14
			1 (last access 2)	3	6
			1 (last access 4)	b	12
12608 0x3140	3	TLB hit PT hit	1 (last access 1)	0	5
			1 (last access 3)	8	14
			1 (last access 5)	3	6
			1 (last access 4)	b	12
49225 0xc040	c	TLB miss PT hit PF	1 (last access 6)	c	15
			1 (last access 3)	8	14
			1 (last access 5)	3	6
			1 (last access 4)	b	12