Midterm

Computer Architecture, Fall 2020

1. (30%)

(a) (5%)

Advantage CP 值, 市場, windows x86, 量產, x86 未來性, PowerPC 成本高

Disadvantage 兼容性

(b)

自己做 CP 值比較高，而且已經有做過手機晶片的經驗。同時也可以脫離 Intel 控制。

(c)

X86 不能直接跑，但可以透過 translation 跑。

iPhone app 從 ISA 的觀點來看是可以直接跑的，但實際上可能還是有一些東西需要調整，但總體來說移植應該不困難。

(d)

所有需要的硬體都整合在一個晶片上。高度整合的優勢，CPU 可以就近取得資源，效能較好。

(e)

同樣的面積可以放進更多 transistors。相同的 design 縮小之後可以比較省電。

(f)

SoC + 5nm

(other reasonable answers are also acceptable, e.g. larger cache, more cores, ...)

2. (30%)

(g)

- highest precision: FP64
- lowest precision: BF16
- lowest range: FP16

(h)

Strike a balance between *performance* and *accuracy*.

- **只寫跑比較快 or 足夠準**: 扣二分
- **只寫因為有 library 支援**: 扣三分

(i)

犧牲一點 accuracy。

- **犧牲效能，因為需要額外的指令做轉換**: 扣三分

  因為轉換後 TF32 的計算速度比 FP32 快，可以提昇整體的效能；而且在設計 (l) 小題的硬體時可以發現，轉換所需的 cost 微乎其微。

(j)

From the viewpoint of A100 vs V100

Not fair. 浮點數格式不同/硬體不同。

However, in the other benchmark (FP16), NVIDIA shows that A100 is still 3x faster than V100.

From the viewpoint of FP32 vs TF32

Not fair. 浮點數格式不同/硬體不同。

針對一些特定應用，不需要那麼準，所以這樣的結果是有意義的。

- **沒有 Defend**: 扣兩分

(k)

Advantages faster

Disadvantages smaller range, lower precision

(l)

FP32 to TF32: drop 13 less significant bits in *precision*.

TF32 to FP16: drop 3 more significant bits in *range*.

- **沒有畫到 sign bit**: 扣一分

3. (40%)

- 有看到兩種setting，data hazard插2個nop，和插3個nop，都給對，再多就不行。但setting在(q),(r)要一樣。會根據(p)題的作答決定你的setting。不一致視情況扣分。
- (p)沒寫過程，扣兩分，跟答案又不符五分；有寫過程，多nop,少nop，多bubble,少bubble，五分往下扣。
- Control hazard一定是2 nop。
- 有要optimize的，reorder後只要沒有多nop或少nop，就全給，不用是最optimized
- Reorder後多寫nop,少寫nop，多bubble,少bubble，扣一分。
- 沒寫過程，先扣3分，寫出來的cycle數不是optimized的，會視cycle數和optimized之差，往下扣。
- Reorder排法影響正確性且用nop救不回來的，扣5分

(m) (5%)

不行。

```
ld x11, 8(x13)
add x12, x10, x11

subi x13, x13, 16
bne x13, LOOP

bne x13, LOOP
XX
XX
```

(n) (5%)

Setting 1

```
ld x10, 0(x13)
ld x11, 8(x13)
nop
nop
add x12, x10, x11
subi x13, x13, 16
nop
nop
bnez x13, LOOP
nop
nop
```

Setting 2

```
ld x10, 0(x13)
ld x11, 8(x13)
nop
nop
nop
add x12, x10, x11
subi x13, x13, 16
nop
nop
nop
bnez x13, LOOP
nop
nop
```

(o) (5%)

subi下的data hazard寫成control hazard，扣一分；沒講到bnez下的control hazard，扣一分

Setting 1

```
ld x10, 0(x13)
ld x11, 8(x13)
nop // data hazard
nop // data hazard
add x12, x10, x11
subi x13, x13, 16
nop // data hazard
nop // data hazard
bnez x13, LOOP
nop // control hazard
nop // control hazard
```

Setting 2

```
ld x10, 0(x13)
ld x11, 8(x13)
nop // data hazard
nop // data hazard
nop // data hazard
add x12, x10, x11
subi x13, x13, 16
nop // data hazard
nop // data hazard
nop // data hazard
bnez x13, LOOP
```

```
nop // control hazard
nop // control hazard
```

(p) (5%)

Setting 1

cycle per iteration: 11 (寫錯就沒了)

Setting 2

cycle per iteration: 13 (寫錯就沒了)

(q) (5%)

Setting 1

```
ld x10, 0(x13)
ld x11, 8(x13)
subi x13, x13, 16
nop                // 可互換
add x12, x10, x11 // 可互換
bnez x13, LOOP
nop
nop
```

8 cycles

Setting 2

```
ld x10, 0(x13)
ld x11, 8(x13)
subi x13, x13, 16
nop                // 可互換
nop                // 可互換
add x12, x10, x11 // 可互換
bnez x13, LOOP
nop
nop
```

9 cycles

(r) (5%)

Yes,

Setting 1

```
ld x10, 0(x13)
ld x11, 8(x13)
bubble
add x12, x10, x11
subi x13, x13, 16
bnez x13, LOOP
bubble
bubble
```

8 cycles

Setting 2

```
ld x10, 0(x13)
ld x11, 8(x13)
bubble
add x12, x10, x11
```

```
subi x13, x13, 16
bnez x13, LOOP
bubble
bubble
```

8 cycles

(s) (5%)

可以

Setting 1

```
ld x10, 0(x13)
ld x11, 8(x13)
add x12, x10, x11
subi x13, x13, 16
bnez x13, LOOP
bubble
bubble
```

speedup: 8/7

Setting 2

```
ld x10, 0(x13)
ld x11, 8(x13)
add x12, x10, x11
subi x13, x13, 16
bnez x13, LOOP
bubble
bubble
```

speedup: 8/7

(t) (5%)

Always taken and implement a branch target buffer

Statistics

Mean	75.07
Max	100
Standard deviation	13.03

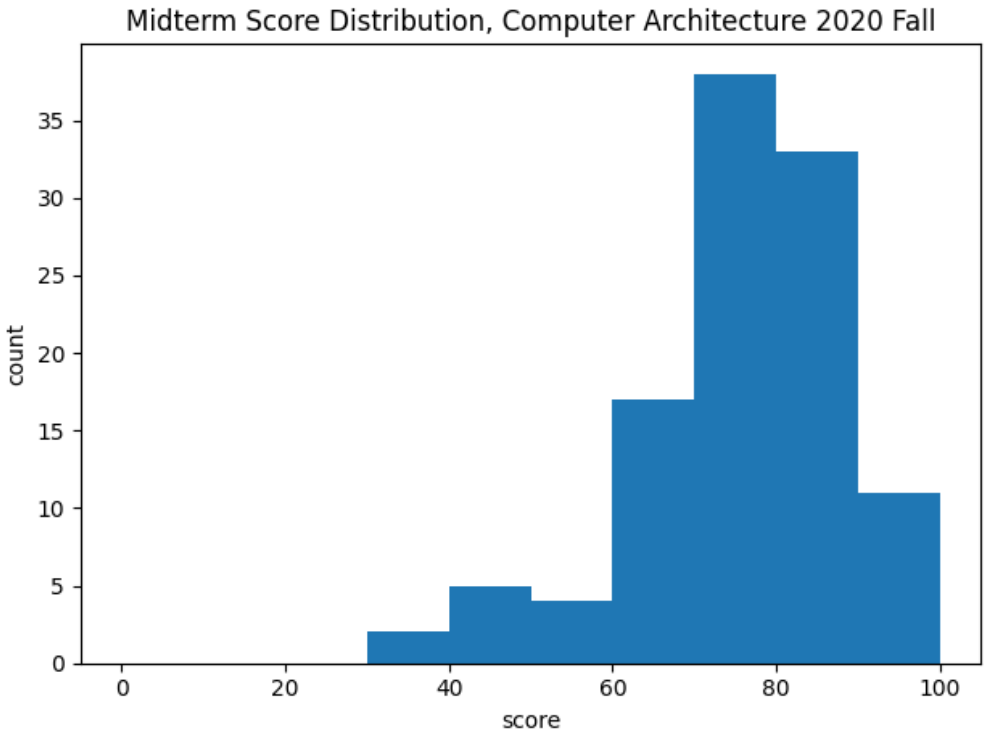

Figure 1: Midterm score distribution (every 10 points)