Lecture 11: Finish ILP

Tuesday, February 13, 2018 10:38 AM

Outline

- Finish ILP from last time
- Review for test

Announcements

A-M come to Wellman 202

N-Z go to Chem 166

1 8.5 x 11 sheet of handwritten notes

Friday: Open lab session w/ Bradley

Last time

```
for (int i=0; i<10000; i++)
  sum += a[i];
```

$1 -> sum
$2 -> &a[i]
$3 -> &a[N+1]

```
loop: lw $4, 0($2)
      add $1, $1, $4
      addi $2, $2, 4
      bne $2, $3, loop(-4)
```

```
for (i=0, i<10000; i+=4)
  sum += a[i]
      += a[i+1]
      += a[i+2]
      += a[i+3]
```

```
loop: lw $4, 0($2)
      add $1, $2, 4
      addi $2, $2, 4
      lw $6, 4($2)
      add $1, $1, $5
      lw
      add $1
      lw
      add $1
      addi $2, $2, 16
      bne
```

RAW

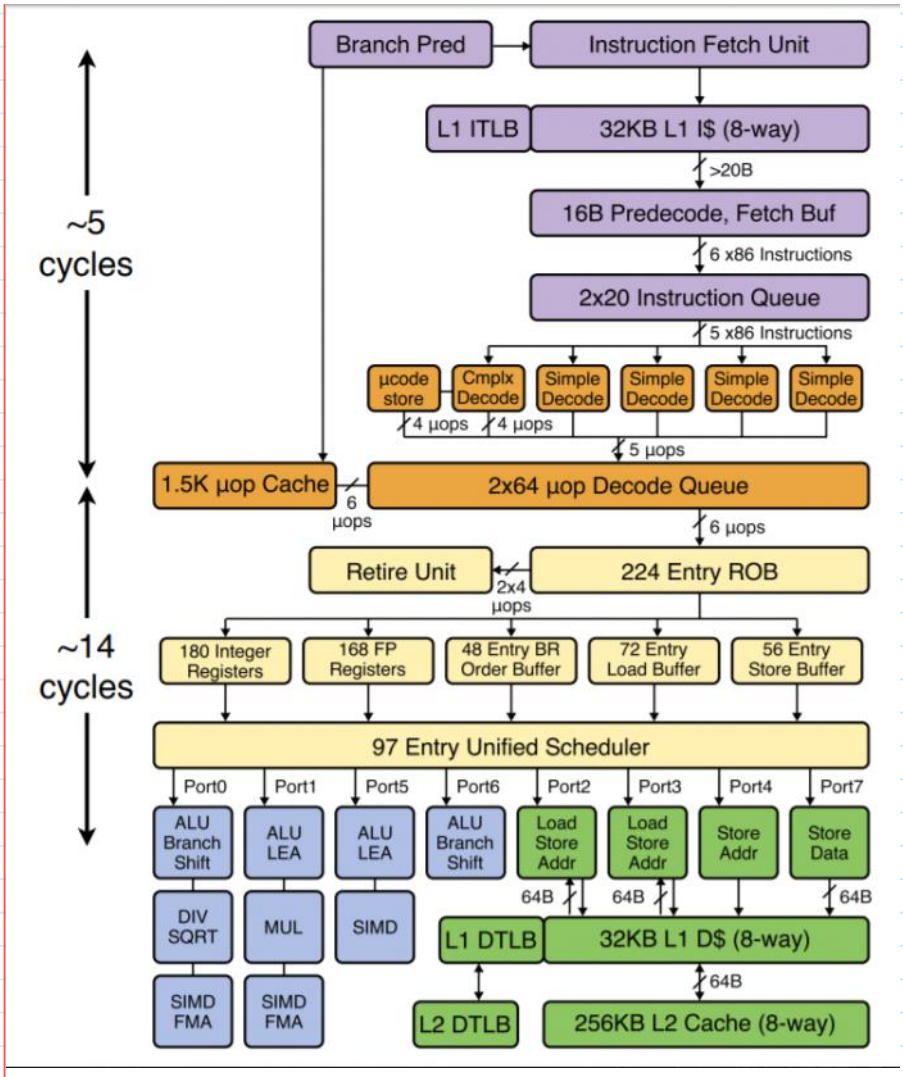
```
lw
lw
lw
lw
add
add
add
add
addi
bne
```

```
Read  -> lw $8, 12($2)
         add $1, $1, $8
writing  addi $2, $2, 16
```

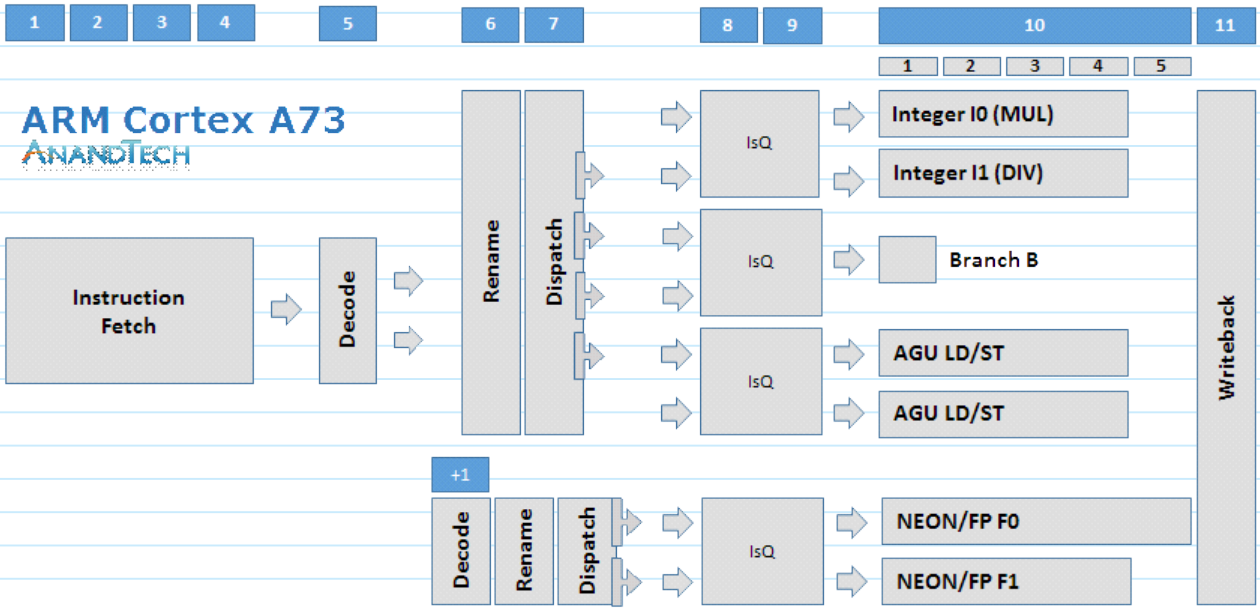
RAW dep

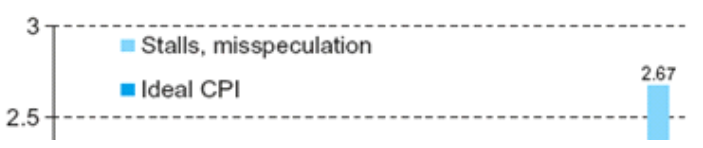
completes after lw

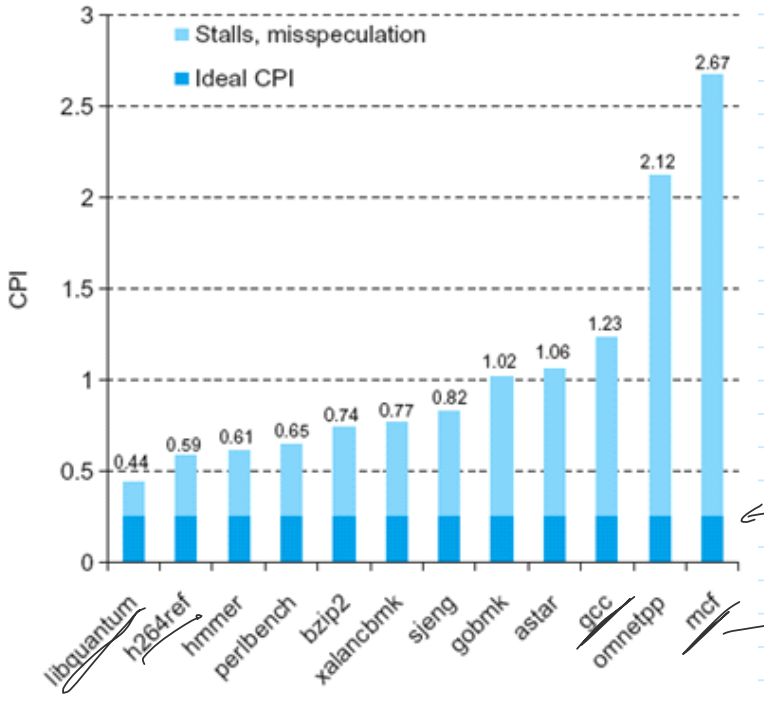
WAR
write after read dependence


~5 cycles
~14 cycles
Branch Pred
Instruction Fetch Unit
L1 ITLB
32KB L1 I$ (8-way)
>20B
16B Predecode, Fetch Buf
6 x86 Instructions
2x20 Instruction Queue
5 x86 Instructions
µcode store
Cmplx Decode
Simple Decode
Simple Decode
Simple Decode
Simple Decode
4 µops
4 µops
5 µops
1.5K µop Cache
6 µops
2x64 µop Decode Queue
6 µops
Retire Unit
2x4 µops
224 Entry ROB
180 Integer Registers
168 FP Registers
48 Entry BR Order Buffer
72 Entry Load Buffer
56 Entry Store Buffer
97 Entry Unified Scheduler
Port0
Port1
Port5
Port6
Port2
Port3
Port4
Port7
ALU Branch Shift
ALU LEA
ALU LEA
ALU Branch Shift
Load Store Addr
Load Store Addr
Store Addr
Store Data
64B
64B
64B
DIV SQRT
MUL
SIMD
L1 DTLB
32KB L1 D$ (8-way)
64B
SIMD FMA
SIMD FMA
L2 DTLB
256KB L2 Cache (8-way)

Arm A73


1
2
3
4
5
6
7
8
9
10
11
1
2
3
4
5
ARM Cortex A73
ANANDTECH
Instruction Fetch
Decode
Rename
Dispatch
IsQ
IsQ
IsQ
Integer I0 (MUL)
Integer I1 (DIV)
Branch B
AGU LD/ST
AGU LD/ST
Writeback
+1
Decode
Rename
Dispatch
IsQ
NEON/FP F0
NEON/FP F1


3
2.5
Stalls, misspeculation
Ideal CPI
2.67



← 0.25 or 4 inst per cycle

→ very memory bound