Fiscal Year 2018



Course number: CSC.T433 School of Computing, Graduate major in Computer Science

# Advanced Computer Architecture

## 10. Instruction Level Parallelism: Out-of-order Execution and Multithreading

www.arch.cs.titech.ac.jp/lecture/ACA/ Room No.W936 Mon 13:20-14:50, Thr 13:20-14:50

Kenji Kise, Department of Computer Science kise \_at\_ c.titech.ac.jp

#### Instruction pipeline of OoO execution processor

- Allocating instructions to instruction window is called dispatch
- Issue or fire wakes up instructions and their executions begin
- In commit stage, the computed values are written back to ROB
- The last stage is called retire or graduate. The result is written back to register file (architectural register file) using a logical register number.

#### In-order front-end



## Register dataflow

• In-flight instructions are ones processing in a processor



#### Case 1: Register dataflow from a far previous instn

- One source operand of insn I2 is from a retired instruction Ia.
- Because Ia is retired, the destination register has no renamed tag. The tag of a source register can not be renamed at renaming stage, still having a logical register tag \$3.
- Where does the operand \$3 comes from?

Ia: add \$3,\$0,\$0
I1: sub p9,\$1,\$2
I2: add p10,p9,\$3
I3: or p11,\$4,\$5
I4: and p12,p10,p11



## Register renaming again

- A processor remembers a set of renamed logical registers.
- If \$1 and \$2 are not renamed for in-flight instructions, it uses \$1 and \$2 instead of p1 and p2.



#### Case 2: Register dataflow from ROB

- Assume that one source operand p10 of insn I5 is from I2 which is not retired. The operand is generated a few clock cycles (tens of cycles sometimes) earlier.
- Because I2 is not retired, RF does not have the operand. Because I2 is committed, the operand is stored in ROB.
- Where does the operand comes from?

Ia: add \$3,\$0,\$0
I1: sub p9,\$1,\$2
I2: add p10,p9,\$3
I3: or p11,\$4,\$5
I4: and p12,p10,p11
I5: nor p13,p10,p12



#### Case 3: Register dataflow from ALUs

- Assume that the other source operand p12 of insn I5 is from I4 which is not committed. The operand is generated in the previous clock cycle.
- Because I2 is not retired, RF does not have the operand.
   Because I2 is not committed, ROB does not have the operand. Ia
- Where does the operand comes from?

Ia: add \$3,\$0,\$0
I1: sub p9,\$1,\$2
I2: add p10,p9,\$3
I3: or p11,\$4,\$5
I4: and p12,p10,p11
I5: nor p13,p10,p12



#### Reorder buffer (ROB)

- Each ROB entry has these fields
  - valid bit, data valid bit, data, target register number
- ROB provides large physical registers for renaming
- A physical register is an item within a matching ROB entry
  - physical register number is ROB entry number



#### Reservation station (RS)

- To simplify the wakeup and select logic, each functional unit (ALU) has own instruction window, an entry for an instruction is called reservation station (RS).
- Each reservation station has
  - valid bit, src1 tag, src1 data, src1 ready, src2 tag, src2 data, src2 ready, dst, operation, ...
  - The computed data with its tag is broadcasted to all RSs.



#### Datapath of OoO execution processor



#### Memory dataflow and branches

- The update of a data cache cannot be recovered easily. So, load and store instructions are executed in-order manner.
  - About 30% (or less) of executed instructions are load and stores.
  - Even if they are executed in-order, IPC of 3 can be achieved.
- Branch instructions are executed in-order manner.
  - About 20% (or less) of executed instructions are jump and branch instructions.
  - Out-or-order branch execution and aggressive miss recovery may cause false recovery (recovery by a branch on the false control path).



#### Pollack's Rule

 Pollack's Rule states that microprocessor "performance increase due to microarchitecture advances is roughly proportional to the square root of the increase in complexity". Complexity in this context means processor logic, i.e. its area.



WIKIPFDIA

## Multithreading (1/2)

- During a branch miss recovery and access to the main memory by a cache miss, ALUs have no jobs to do and have to be idle.
- Executing multiple independent threads (programs) will mitigate the overhead.
- They are called coarse- and fine-grained multithreaded processors having multiple architecture states.



# Multithreading (2/2)

• Simultaneous Multithreading (SMT) can improve hardware resource usage.





#### From multi-core era to many-core era



Figure 1: Current and expected eras of Intel® processor architectures

#### Platform 2015: Intel® Processor and Platform Evolution for the Next Decade, 2005

#### Homework 6

- Design a register renaming unit which renames two instructions per cycle in Verilog HDL. Please download rename01.v and rename02.v from the support page and refer them.
- 2. Please modify a module RENAME in rename02.v referring the design which renames one instruction per cycle in rename01.v.
  - The renamed instruction sequences by rename01.v and rename02.v must be the same.
- 3. Submit a report printed on A4 paper at the beginning of the next lecture.
  - The report should include a block diagram, a source code in Verilog HDL, and obtained waveforms of your design.

0:	add	\$00,	\$00,	\$00	->	add	p08,	p03,	p03
1:	add	\$05,	\$01,	\$02	->	add	p09,	р03,	p03
2:	add	\$09,	\$05,	\$04	->	add	p10,	p09,	p03
3:	add	\$05,	\$05,	\$02	->	add	p11,	p09,	p03
4:	add	\$05,	\$06,	\$07	->	add	p12,	р03,	p03
5:	add	\$05,	\$05,	\$05	->	add	p13,	p12,	p12
6:	add	\$05,	\$05,	\$07	->	add	p14,	p13,	p03
7:	add	\$05,	\$05,	\$09	->	add	p15,	p14,	p10
8:	add	\$13,	\$11,	\$12	->	add	p16,	р03,	p03
9:	add	\$14,	\$11,	\$12	->	add	p17,	р03,	p03
10:	add	\$15,	\$11,	\$12	->	add	p18,	р03,	p03
11:	add	\$13,	\$13,	\$00	->	add	p19,	p16,	p03
12:	add	\$14,	\$14,	\$00	->	add	p20,	p17,	p03
13:	add	\$15,	\$15,	\$00	->	add	p21,	p18,	p03
14:	add	\$16,	\$14,	\$15	->	add	p22,	p20,	p21
15:	add	\$17,	\$16,	\$13	->	add	p23,	p22,	p19
16:	add	\$00,	\$00,	\$00	->	add	p24,	p03,	p03
17:	add	\$00,	\$00,	\$00	->	add	p25,	p03,	p03



#### Renaming two instructions per cycle for superscalar

Renaming instruction IO and I1

