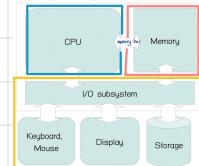


COM ARCH

★ 1) Introduction ★ Com arch គឺជាពេលវត្ថុទាំងអស់នៃបច្ចេកទេស, Transf CPU, និមួយនាក់នូវខ្លួន, ISA, ភាគខ្លួន memory, និងចំណែកគ្នាបែងចែង



- 1) CPU
- 2) memory
- 3) I/O system

ឯកសារក្នុងការគ្រប់គ្រងបច្ចេកទេស, CPU instruction set (ISA), Instruction Set Architecture → ទំនើបស្ថិកបច្ចេកទេសដោយប្រើប្រាស់CPU
ឯកសារក្នុងការគ្រប់គ្រងបច្ចេកទេស, CPU instruction set (ISA) → Instruction-level compatibility (Instruction-level compatibility)
2) ទំនើបក្នុងគម្រោង (Computer Organization) → ទំនើបក្នុងគម្រោងនៃបច្ចេកទេស CPU & ឧបតាថ្មី សារការការពារ
ឯកសារក្នុងការគ្រប់គ្រង, CPU → គឺជាមុន ALU, control unit, register, bus (wire/wirebus)
3) ការអនុវត្តន៍ (Implementation) ការអនុវត្តន៍ និងការអនុវត្តន៍ នឹងបានធ្វើឡើងនៅក្នុងបច្ចេកទេស
* ពីរបានបានបញ្ជាផី និងបានបញ្ជាផី និងបានបញ្ជាផី

សារការការពារ

↓ compiler

Intermediate / Assembly

↓ Assembler

Binary / Machine Code

★ 2) Performance ★

$$P = \frac{1}{T}$$

រាយការណ៍ P → if X ទៅការ / P1 នៃ Y ពេល n ឱ្យ speed up

T = Execution Time

P = Performance

$$\text{Speedup (n)} = \frac{P_x (1/\text{sec})}{P_y (1/\text{sec})} = \frac{\frac{1}{T_x}}{\frac{1}{T_y}} = \frac{T_y}{T_x}$$

នូវការបង្ហាញ P

① ការពារ នូវការបង្ហាញ

② ពេល f នូវ clock

③ ការពារ ការពារ

• ពារាយការណ៍ (response time) & រាយការណ៍ការពារ (throughput) → សារការការពារ និងបច្ចេកទេស (aka នូវការការពារ និងបច្ចេកទេស)

• ឯកសារក្នុងការពារ (Benchmark) • SPEC Benchmark Suites → benchmark នឹង SPEC System Performance Evaluation Corporation ការពារ និងបច្ចេកទេស

• Real benchmark: នូវការការពារ និងបច្ចេកទេស នៃការពារ និងបច្ចេកទេស P • Microbenchmark: នូវការការពារ និងបច្ចេកទេស P • Kernel: នូវការការពារ និងបច្ចេកទេស hardware

• Synthetic benchmark: នូវការការពារ និងបច្ចេកទេស នៃការពារ និងបច្ចេកទេស

• ឯកសារក្នុងការពារ P នៃ 1. ពារាយការណ៍ និងបច្ចេកទេស 2. រាយការណ៍ hardware និងបច្ចេកទេស 3. ឯកសារការពារនៃការពារ និងបច្ចេកទេស 4. ឯកសារការពារនៃការពារ និងបច្ចេកទេស

• ឯកសារក្នុងការពារ P → ឯកសារក្នុងការពារ Speed up (តាមការពារ និងបច្ចេកទេស)

↳ Geometric Mean $\sqrt[n]{T_X}$; (ឯកសារក្នុងការពារ និងបច្ចេកទេស)

* * * Amdahl's Law overall = $\frac{T_{\text{total}}}{T_{\text{new}}}$

Overall Speedup = $\frac{1}{(1-P) + \frac{P}{n}}$

P: សារការណ៍ការពារ និងបច្ចេកទេស

n: បច្ចេកទេស និងបច្ចេកទេស (នៅក្នុង)

$$\text{CPU time} = \frac{1}{\text{Cycle time (sec)}} = \frac{1}{\text{Clock Rate (Hz)}}$$

$$\text{CPI average} = \sum_{i=1}^n \text{CPI}_i \times \text{Weight}_i; \text{ CPI} = \text{Cycle per instruction}$$

CPU time = CPI average × Cycle time seconds = $\frac{[\text{Clock Cycle}]}{\text{Instruction}} \times \text{Seconds per cycle}$

CPU Time = Instruction Count × CPU time/Instruction = Instruction × CPI average × Cycle time

★ 3.) ISA និងការគ្រប់គ្រង ★ Instructions / Operators → ការពារ → Opcode, Operands → ការពារការពារ និងបច្ចេកទេស, Register និងការពារនៃការពារ CPU \$, Immediate ← n #

CISC = Complex Instruction Set Computing → memory-memory RISC = Reduce Instruction Set Computing → register-register

• សមតាមរាយការ Accumulator → នូវការពារនៃការពារ និងបច្ចេកទេស

• Memory-Memory → នូវការពារ និងបច្ចេកទេស និងការពារ និងបច្ចេកទេស (គ្រប់គ្រង)

• Stack → operands និងការពារ stack និងបច្ចេកទេស និងការពារ និងបច្ចេកទេស (គ្រប់គ្រង)

• Register-Register / load-store machine → operands និងការពារ និងបច្ចេកទេស និងការពារ និងបច្ចេកទេស (គ្រប់គ្រង)

• សមតាមការពារ & រាយការណ៍ → 1. ពារាយការណ៍ និងបច្ចេកទេស 2. ពេលវត្ថុ និងបច្ចេកទេស 3. នូវការការពារ និងបច្ចេកទេស 4. នូវការការពារ និងបច្ចេកទេស

• នូវការការពារ និងបច្ចេកទេស 1. Endian → big endian : នូវការការពារ និងបច្ចេកទេស 2. little endian : នូវការការពារ និងបច្ចេកទេស

• នូវការការពារ និងបច្ចេកទេស 1. Memory Alignment : នូវការការពារ និងបច្ចេកទេស និងការពារ និងបច្ចេកទេស និងការពារ និងបច្ចេកទេស (គ្រប់គ្រង)

• Arch និងការពារ CPU និងបច្ចេកទេស 1. ex! ARM (non. mmr)

• ឯកសារក្នុងការពារ & Exception

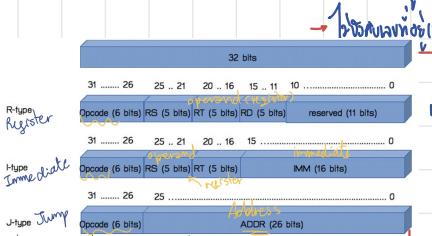
• ឯកសារក្នុងការពារ & Exception 1. ពារាយការណ៍ register 2. ពេលវត្ថុ memory

• ឯកសារក្នុងការពារ local & register → 1. caller save 2. callee save

• ឯកសារក្នុងការពារ local & register → 1. caller save 2. callee save (software) (hardware)

• Exception & Interrupt → សែនបោន្ទាល់ → vector table, handler table, fix entry

• ឯកសារក្នុងការពារ single cycle



Addressing Mode	តាមរយៈនាំ នៃការពារ និងបច្ចេកទេស	RTL	ប្រើប្រាស់ការពារ
1. Immediate	LDI \$r1, #4	\$r1 ← 4	A = 4
2. Register Direct	ADD \$r1, \$r2, \$r3	\$r1 ← \$r2 + \$r3	A = B + C
3. Displacement	LD \$r1, 100(\$r2)	\$r1 ← MEM[\$r2 + 100]	A = D[i+100] និង A = Gage
4. Register Indirect	LD \$r1, (\$r2)	\$r1 ← MEM[\$r2]	A = *p
5. Index	LD \$r1, (\$r2 + \$r3)	\$r1 ← MEM[\$r2 + \$r3]	A = D[i + \$r3] និង A = Gage
6. Direct	LD \$r1, 1000	\$r1 ← MEM[1000]	
7. Memory Indirect	LD \$r1, @(\$r2)	\$r1 ← MEM[MEM[\$r2]]	
8. Autoincrement	LD \$r1, (\$r2)+	\$r1 ← MEM[\$r2]	A = D[i+1]
9. AutoDecrement	LD \$r1, (\$r2)-	\$r1 ← MEM[\$r2]	A = D[i-1]
10. Scale	LD \$r1, (\$r2)(\$r3*4)	\$r1 ← MEM[\$r2 + \$r3*4]	F = E[i]

• ឯកសារក្នុងការពារ & Exception

• ឯកសារក្នុងការពារ & Exception 1. ពារាយការណ៍ register 2. ពេលវត្ថុ memory

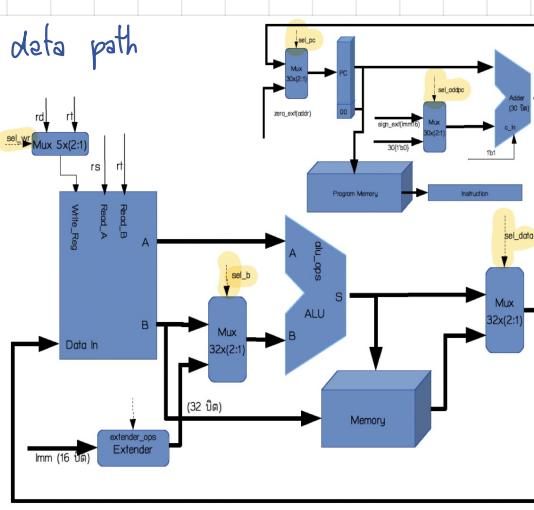
• ឯកសារក្នុងការពារ local & register → 1. caller save 2. callee save

• ឯកសារក្នុងការពារ local & register → 1. caller save 2. callee save (software) (hardware)

• Exception & Interrupt → សែនបោន្ទាល់ → vector table, handler table, fix entry

• ឯកសារក្នុងការពារ single cycle

ORI rt, rs, imm	: (I-type) opcode = 010000
គារអនុវត្តន៍ R(rt) ← R(rs) zero_ext(imm);	PC ← PC + 4
ORUI rt, rs, imm	010001
គារអនុវត្តន៍ R(rt) ← R(rs) zero_pad(imm);	PC ← PC + 4
ADD rd, rs, rt	000001
គារអនុវត្តន៍ R(rd) ← R(rs) + R(rt);	PC ← PC + 4
LW rt, imm(rs)	011000
Load គារអនុវត្តន៍ R(rt) ← MEM[R(rs) + sign_ext(imm)];	PC ← PC + 4
SW rt, imm(rs)	011100
Store គារអនុវត្តន៍ MEM[R(rt) + sign_ext(imm)] ← R(rt);	PC ← PC + 4
BEQ rs, rt, imm*4	100100
គារអនុវត្តន៍ If (R(rs) == R(rt)) then PC ← PC + 4 + (sign_ext(imm) * 4)	
else	
PC ← PC + 4	
JMP addr*4	110000
គារអនុវត្តន៍ PC ← (addr * 4)	



Opcode	Zero	sel_pc	sel_dpc	sel_wr	sel_b	sel_data	reg_wr	mem_wr	alu_ops
ORI 010000	x	0	0	1	1	0	1	0	01
ORUI 010001	x	0	0	1	1	0	1	0	01
ADD 000001	x	0	0	0	0	0	0	1	00
LW 011000	x	0	0	1	1	1	1	0	00
SW 011100	x	0	0	x	1	x	0	1	00
BEQ 100100	0	0	0	1	x	x	x	0	10
JMP 110000	x	1	x	x	x	x	x	0	00

* អ្នកអនុវត្តន៍នៅក្នុង 1 clock *

: clock rate = $\frac{1}{\text{នាក់រាយការនៃការពារ}}$

