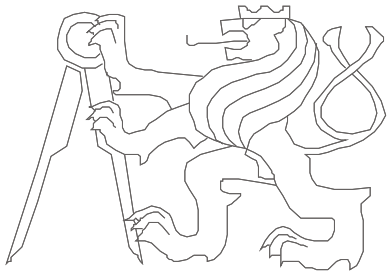
Architektura počítačů

Vývoj architektur mikroprocesorů - od 4 bitů k superskalárnímu RISC

Pavel Píša, Michal Štepanovský



České vysoké učení technické, Fakulta elektrotechnická

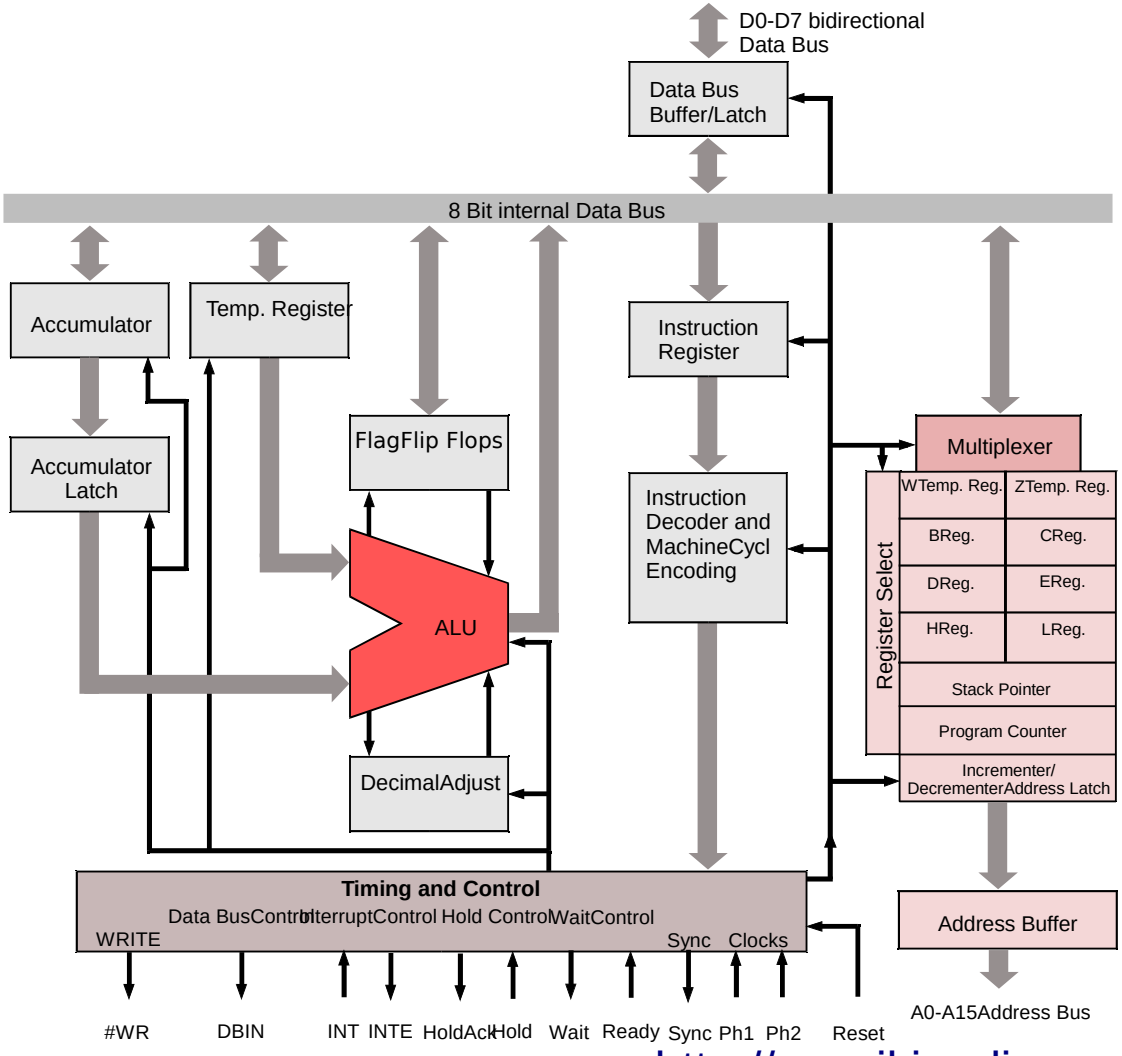
Technologické porovnání

CPU	Výrobce	Rok	Tranzistorů	Techno.	Reg/Bus	Data/prog+IO	Cache I/D+L2	Float	Frekv.	MIPS	Cena
4004	Intel	1971	2,300	10um - 3x4mm	4bit	1kB/4kB			750kHz	0.06	$200
8008	Intel	1972	3,500	10um	8bit	16kB				0.06	
8080	Intel	1974	6,000	6um	8bit	64kB+256			2MHz	0.64	$150
MC6501	NMOS T.	1975									$20
8085	Intel	1976	6,500	3um	8bit	64kB+256			5MHz	0.37	
Z-80	Zilog	1976			8bit	64kB+256			2.5MHz		
MC6502	NMOS T.	1976									$25
8086	Intel	1978	29,000	3um	16/16bit	1MB+64kB			4.77MHz	0.33	$360
8088	Intel	1979		3um	16/8bit	1MB+64kB			4.77MHz	0.33	
MC68000	Motorola	1979	68,000		16-32/16bit	16MB					
80286	Intel	1982	134,000	1.5um	16/16bit	16MB/1GB virt	256B/0B		6MHz	0.9	$380
MC68020	Motorola	1984	190,000		32/32bit	16MB	Ano		16MHz		
80386DX	Intel	1985	275,000	1.5um	32/32bit	4GB/64TB virt			16MHz		$299
MC68030	Motorola	1987	273,000			4GB+MMU	256B/256B				
80486	Intel	1989	1.2mil	1um	32/32bit	4GB/64TB virt	8kB	Ano	25MHz	20	$900
MC68040	Motorola	1989	1.2mil			4GB+MMU	4kB/4kB	Ano			
PowerPC 601	Mot+IBM	1992	2.8mil		32/64bit	2^{56}	32kB	Ano	66MHz		
PA-RISC	HP	1992							50MHz		
Pentium	Intel	1993	3.1mil	0.8um - BiCMOS	32/64bit	4GB+MMU		Ano	66MHz	112	
Alpha	DEC	1994	9.3mil		64bit	4GB/64TB vir	8/8+96kB		300MHz	1000	
MC68060	Motorola	1994	2.5mil			4GB+MMU	8kB/8kB	Ano	50MHz	100	$308
Pentium Pro	Intel	1995	5.5mil					Ano	200/60MHz	440	$1682
Pentium II	Intel	1998	7.5mil		32/64bit			Ano+MMX	400/100MHz	832	
PowerPC G4 MPC7400	Motorola	1999		0.15um - cooper6LM CMOS	64/128bit	4GB/2^{52}	32kB/32kB+2MB	Ano+AV	450MHz	825	

Akumulátorové architektury

- registr+akumulátor → akumulátor
  - 4bit Intel 4004 (1971)
  - 8bit Intel8080 (1974) - adresace s využitím dvojic registrů
  - pouze základní aritmetickologické operace – sčítání, odčítání, posuny pro 8-bit akumulátor
  - volání CALL a RET s ukládáním 16-bit PC na zásobník
  - několik 16-bit operací – inkrement/dekrement dvojic registrů, přičtení k HL a ukládání na zásobník
  - výkon instrukcí řízený mikrokódem – na jednu instrukci 2 až 11 cyklů při 2 MHz hodinách

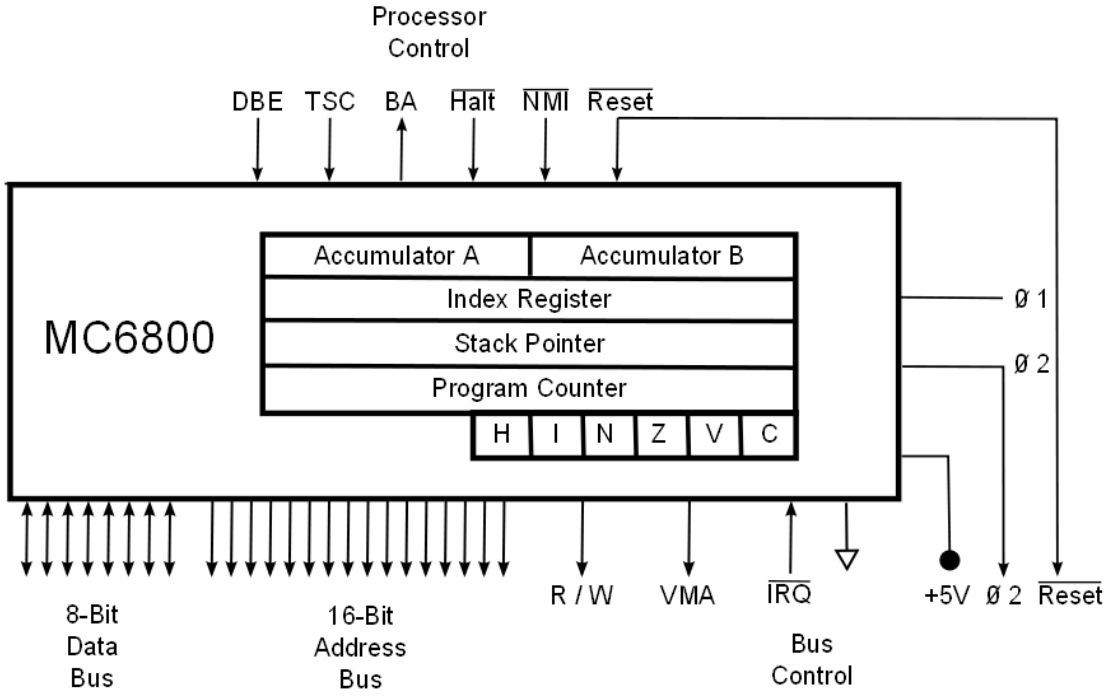
Intel 8080



http://en.wikipedia.org/wiki/Intel_8080

Rychlé paměti ⇒ málo registrů a více adresačních módů

- Motorola 6800, NMOS T. 6502 (1975) - jen akumulátor, index, SP a PC. - využití nulté stránky, HW dekodér
- Texas TMS990 pouze workspace pointer, PC, SP, registry v paměti, podobné u transputerů

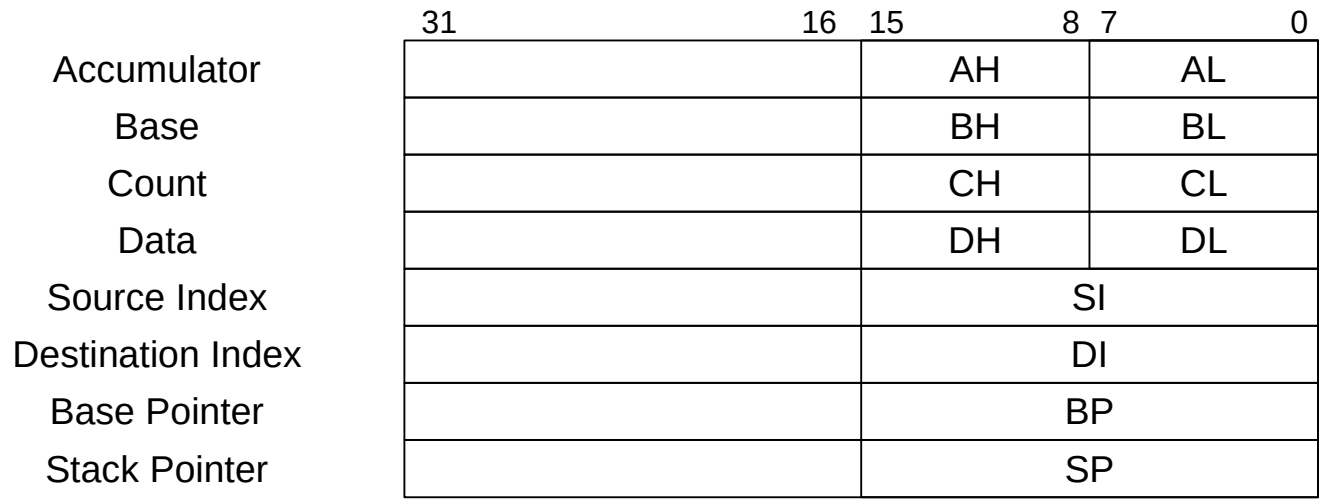


Nedostatečná rychlost paměti ⇒ výkonný komplexní instrukční soubor, blízký C, CISC

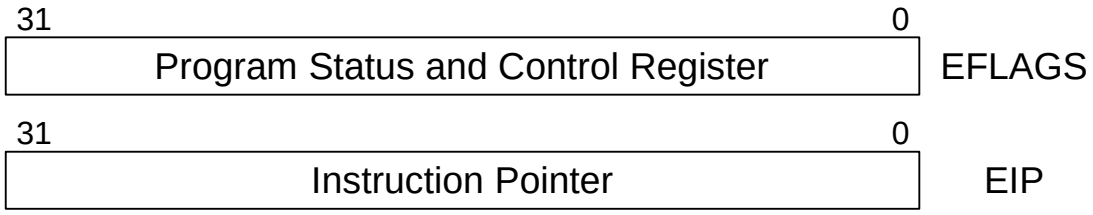
- Intel 8086
- Motorola 68000 (1979) – 16/32bit
  - dvou-operandové instrukce
  - operace registr+=registr, paměť+=registr, registr+=paměť, paměť+=paměť
  - rozsáhlá instrukční sada, nutnost použít mikrokód
- Z-8000 16bit, Z-80000 32bit (1986) CISC
  - 6 fází zpracování, bez mikrokódu, 18000 tranzistorů

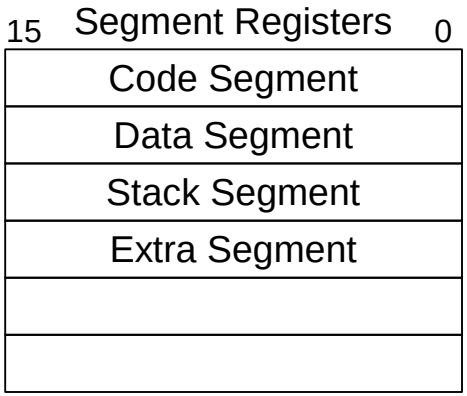
Intel 8086 a 32-bit i386

General-Purpose Registers



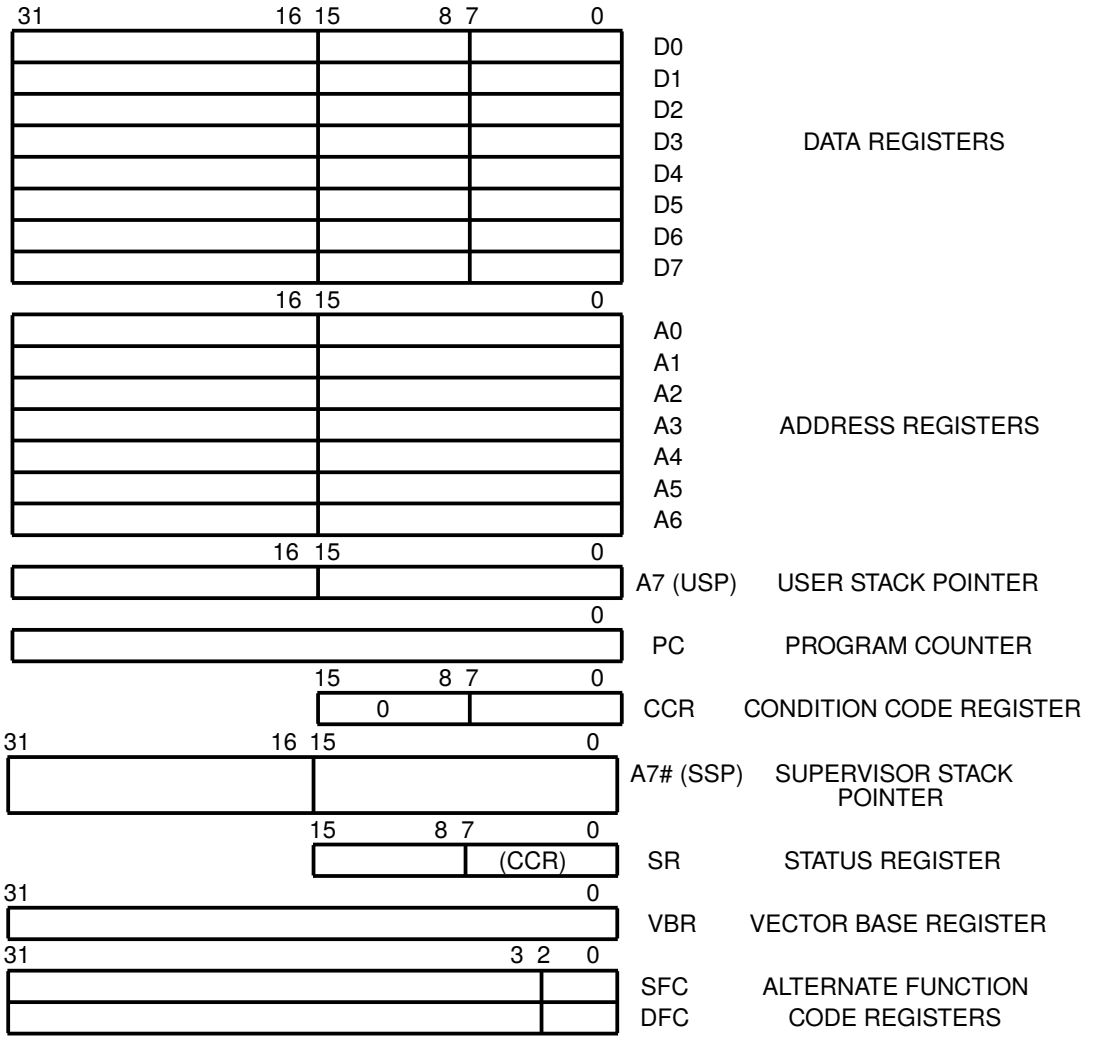
	16-bit	32-bit
Accumulator	AX	EAX
Base	BX	EBX
Count	CX	ECX
Data	DX	EDX
Source Index		ESI
Destination Index		EDI
Base Pointer		EBP
Stack Pointer		ESP





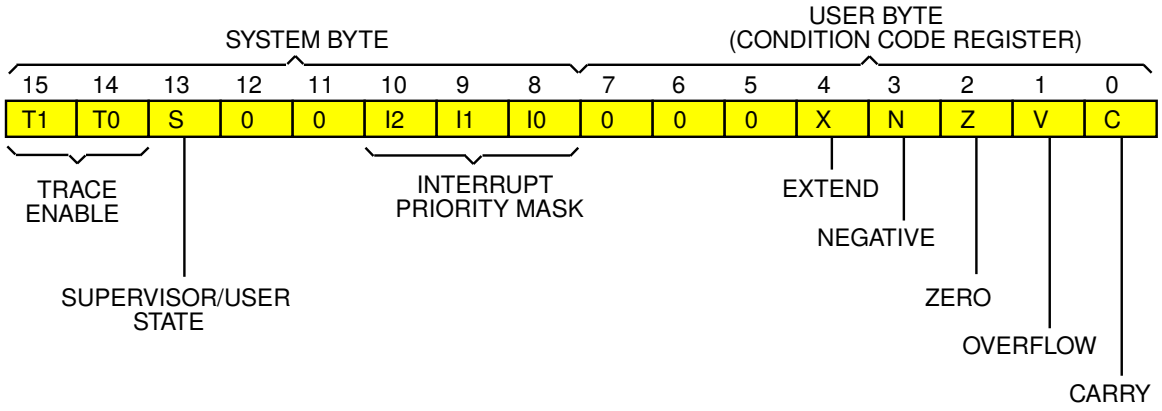
Základní registry procesorů M68xxx/CPU32/ColdFire

Uživatelský model architektury



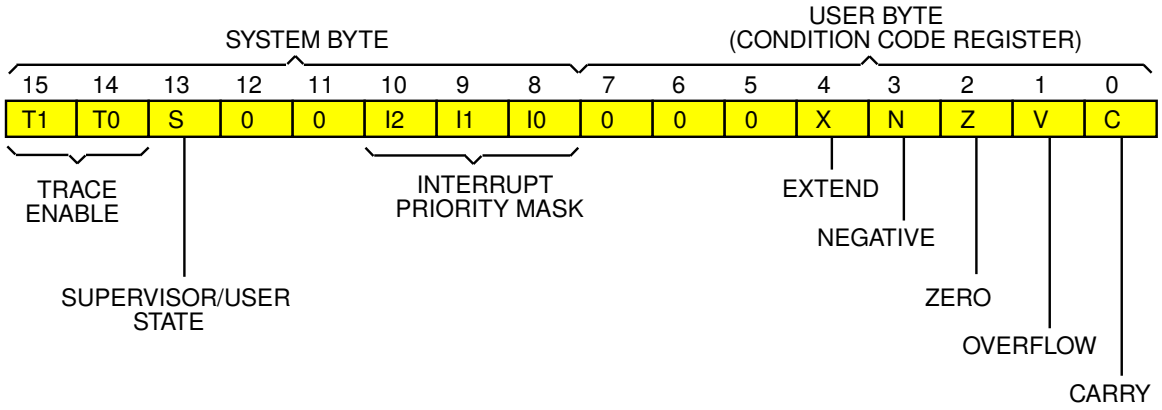
Systémový model

Stavový registr – podmínková část



- N – negative ... = 1 pokud je nejvíce významný bit operandu nebo výsledku roven 1 (negativní výsledek v doplňkovém kódu)
- Z – zero ... = 1 pokud jsou veškeré bity operandu nebo výsledku nulové
- V – overflow .. = 1 pokud je výsledek mimo zobrazitelný rozsah (např. přetečení při znaménkových operacích)
- C – carry ... = 1 pokud dochází k přenosu (carry) z nejvíce významného bitu během aritmetické operace nebo je potřeba výpůjčka (borrow) při odčítání
- X - extend (extended carry) ..použitý při rozšíření operací na vícenásobnou přesnost (odpovídá hodnotě C)

Stavový registr – systémová část



- T1, T0 – trace … pokud je některý z bitů nastavený, dojde ke generování výjimky po provedení každé instrukce nebo po instrukcích změny toku programu (skok, návrat, volání)
- S – supervisor … pokud je nastaven na 1, procesor se nachází v systémovém režimu a SP odpovídá SSP. Jinak se procesor nachází v uživatelském režimu, SP odpovídá USP a manipulace se systémovou částí není možná a na přístupy k paměti (MMU) jsou aplikovaná omezení příslušná uživatelskému režimu
- I2, I1, I0 - interrupt mask … definuje úroveň přerušení po kterou je přijetí žádosti blokované – odložené na později. Výjimkou jsou žádosti úrovně 7, která nejsou maskovatelná

Adresní režimy 68000, 68020, 68040

- Pro přístup k operandům slouží 14 režimů adresace.
- **Rn** obsah datového nebo adresového registru
- **(An)** obsah paměti na adrese **An**
- **(An)+** obsah paměti na adrese **An** s následnou inkrementací registru o hodnotu danou délkou operandu
- **-(An)** nejdříve dojde k dekrementaci registru o délku operandu a pak je registr použit k adresaci
- **(d16,An)** adresový registr s 16 bitovým znaménkovým posunutím
- **(d8,An,Xn)** adresový registr s 8 bitovým znaménkovým posunutím a přičtením indexového registru (případně jen jeho nižších 16 bitů), pro procesory CPU32 a 68020+ může být index násoben číslem 1, 2, 4 nebo 8
- **(xxx).W** 16 bitová absolutní adresa
- **(xxx).L** 32 bitová absolutní adresa

Nízká průchodnost dat a načítání instrukcí ⇒ cache paměti

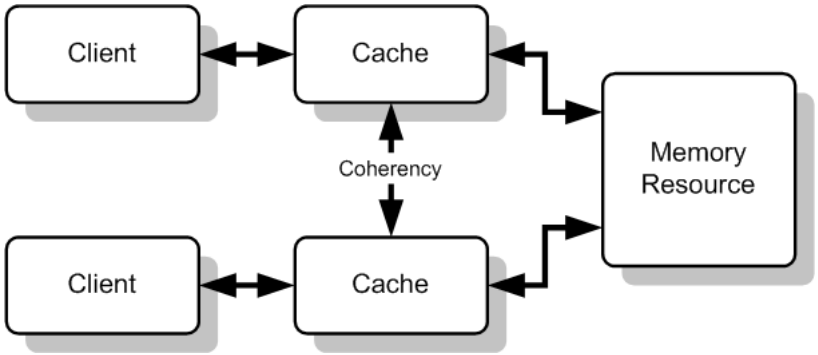
- Celkem dobře řešitelný problém
- Společná cache nebo harvardská I & D
- Další úrovně (rychlost limitovaná pro větší kapacity – dekodér, kapacita společných vodičů)
- Komplikuje se však koherence dat při přístupu přes DMA a pro SMP
  - speciální instrukce pro přístup k periferiím a synchronizaci např. `eieio` (PowerPC), `mcr p15` (ARM), …
  - hardwarová podpora
    - protocol MSI , MESI (Pentium), MOSI
    - MOESI AMD64 (Modified, Owned, Exclusive, Shared, and Invalid)

Koherence dat a vyrovnávací paměti - cache

MOESI protokol

- Modified – řádka cache obsahuje aktuální a modifikovaná data, ostatní CPU s daty nepracují, v paměti jsou data stará
- Owned – řádka obsahuje aktuální data, ta mohou být sdílena i jinými CPU, ale pouze v S, v paměti může být stále předchozí hodnota
- Exclusive – pouze tento CPU a paměť obsahuje data
- Shared – řádka je sdílena i s jinými CPU, pokud některé z nich v O, tak mohou být data rozdílná od starého obsahu hlavní paměti
- Invalid – v řádce nejdou žádná platná data

	M	O	E	S	I
M	N	N	N	N	Y
O	N	N	N	Y	Y
E	N	N	N	N	Y
S	N	Y	N	Y	Y
I	Y	Y	Y	Y	Y



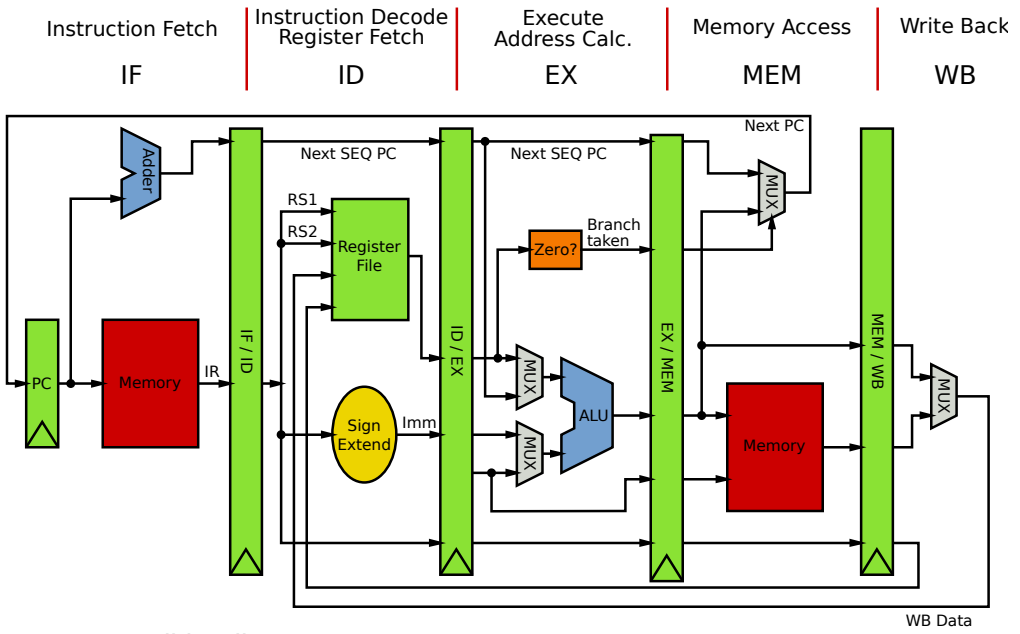
http://en.wikipedia.org/wiki/MOESI_protocol

Zrychlení výkonu instrukcí ⇒ architektury RISC

- Menší vzájemná závislost, tříoperandové instrukce, spekulativní provádění instrukcí, přejmenovávání registrů, eliminace závislosti na příznakovém registru (DEC Alpha, několik př. registrů PowerPC, potlačení nastavování příznaků ARM)
- architektury load-store, operace pouze registr+=registr nebo registr=registr+registr a separátní load-store instrukce.
- Pevná délka instrukčního kódů ⇒ dlouhé programy, ale velmi rychlé dekódování instrukcí, optimalizované pro zřetězené zpracování (pipline)

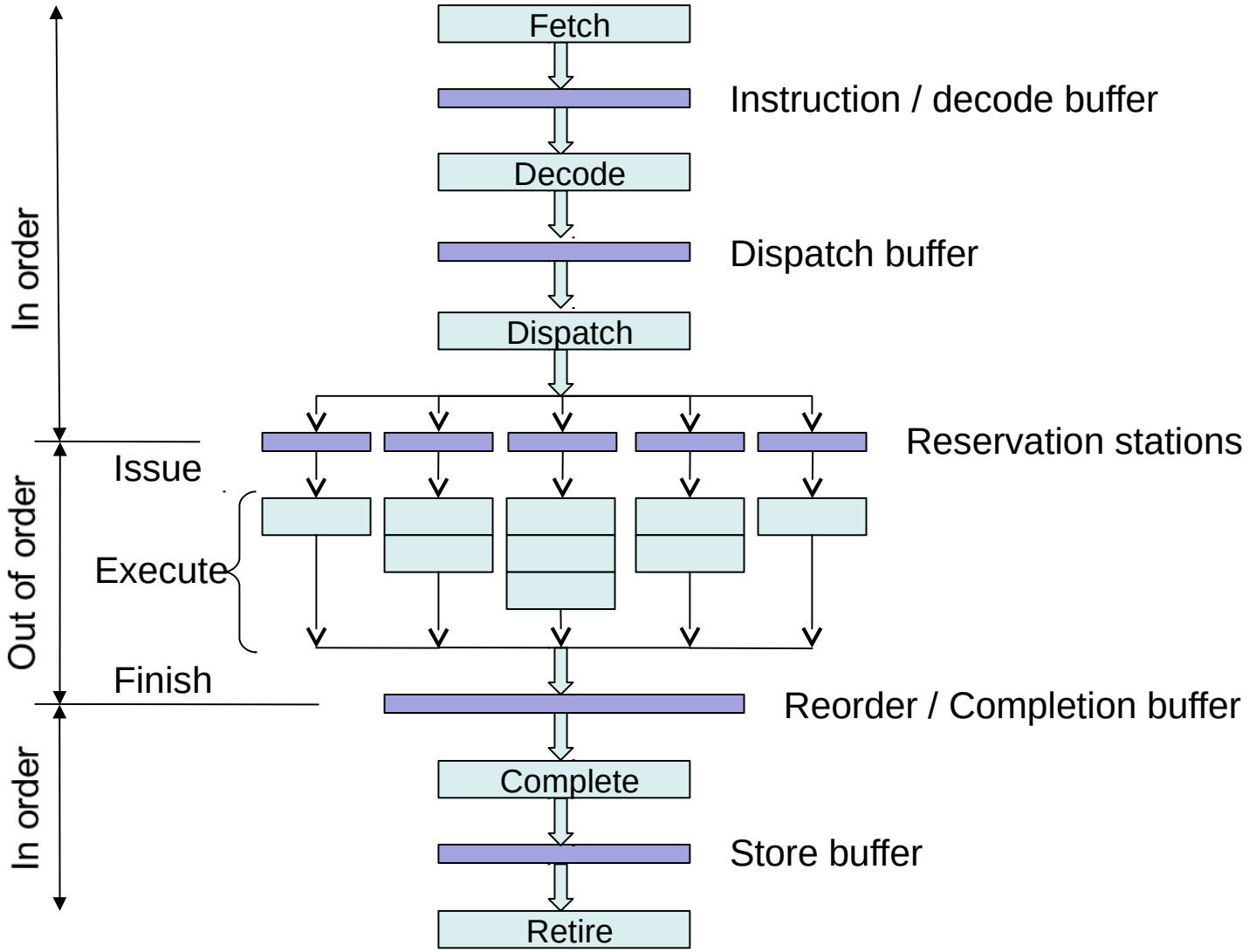
Vícefázové zpracování bez mikrokódu, stále potíže se skoky

- Včasné dekódování skoků
- Využití delay slotů MIPS, DSP
- Statická a dynamická predikce skoků, cache cílů skoků, spekulativní provádění instrukcí



Source: wikipedia

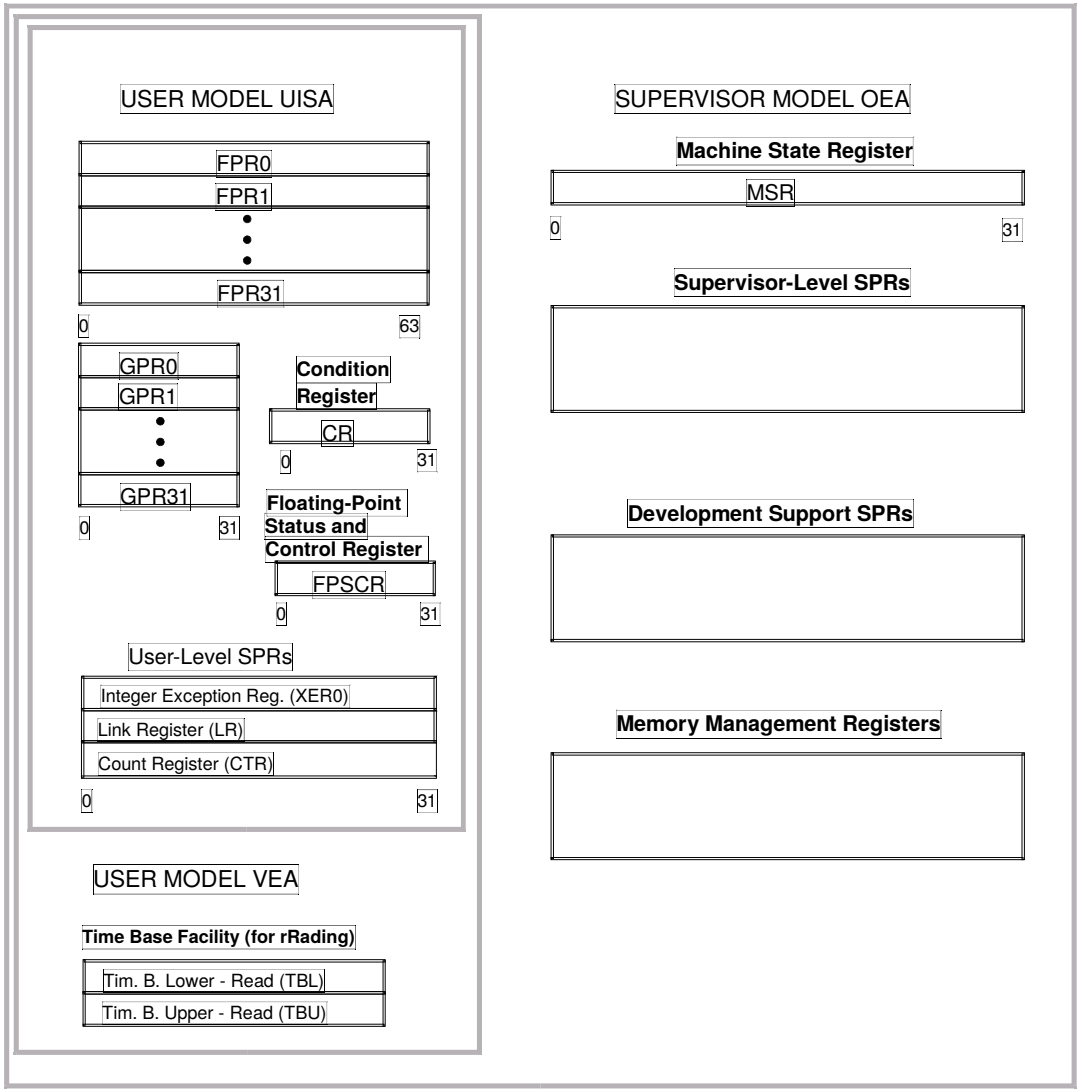
Paralelní zpracování instrukcí – superskalární CPU



Další eliminace přístupů do paměti ⇒ registrová okna, link/return registr

- SPARC - 8 globálních registrů, 8 z předchozího okna (parametry), 16 z aktuálního okna, více jak 100 registrů pro posuny okna. 8 registrů z aktuálního slouží k předání parametrů volané funkci
- PowerPC - zrychlení volání leaf-node funkcí s využitím specializovaného registru (link registr) pro návratovou hodnotu čítače instrukcí

Architektura PowerPC

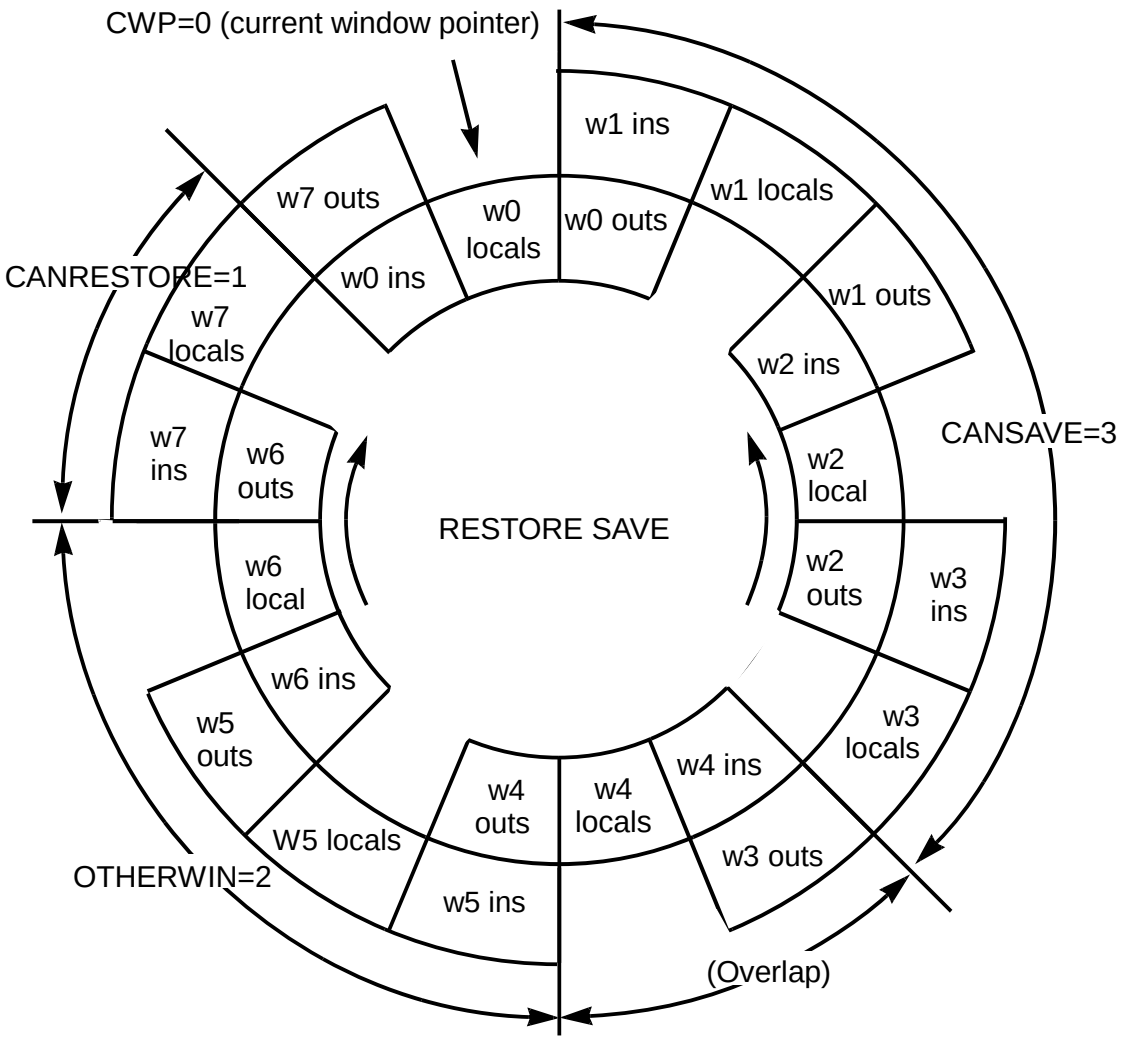


SPARC – registrová okna

- CPU obsahuje 40 až 520 obecně použitelných 32-bitových registrů
- 8 z nich jsou globální registry ostatní jsou po 16 rozdělený do 2 až 32 registrových oken
- Každá instrukce může přistupovat k 8 globálním registrům a 24 registrům v právě přístupném okně
- 24 registrů v okně je rozděleno na 8 vstupních (in), 8 lokálních (local) a 8 registrů ze sousedního okna adresovatelných z aktuálního okna jako výstupní (out) registry
- Aktivní okno je dané hodnotou 5-bitového ukazatele Current Window Pointer (CWP).
- CWP je dekrementován při vstupu do podprogramu a další okno je nastaveno na aktivní
- Inkrementace CWP navrátí předchozí okno
- Bitový registr Window Invalid Mask (WIM) umožní označit libovolná okna za neplatná a vyžádat při jejich aktivací výjimku (overflow or underflow)

SPARC - registry

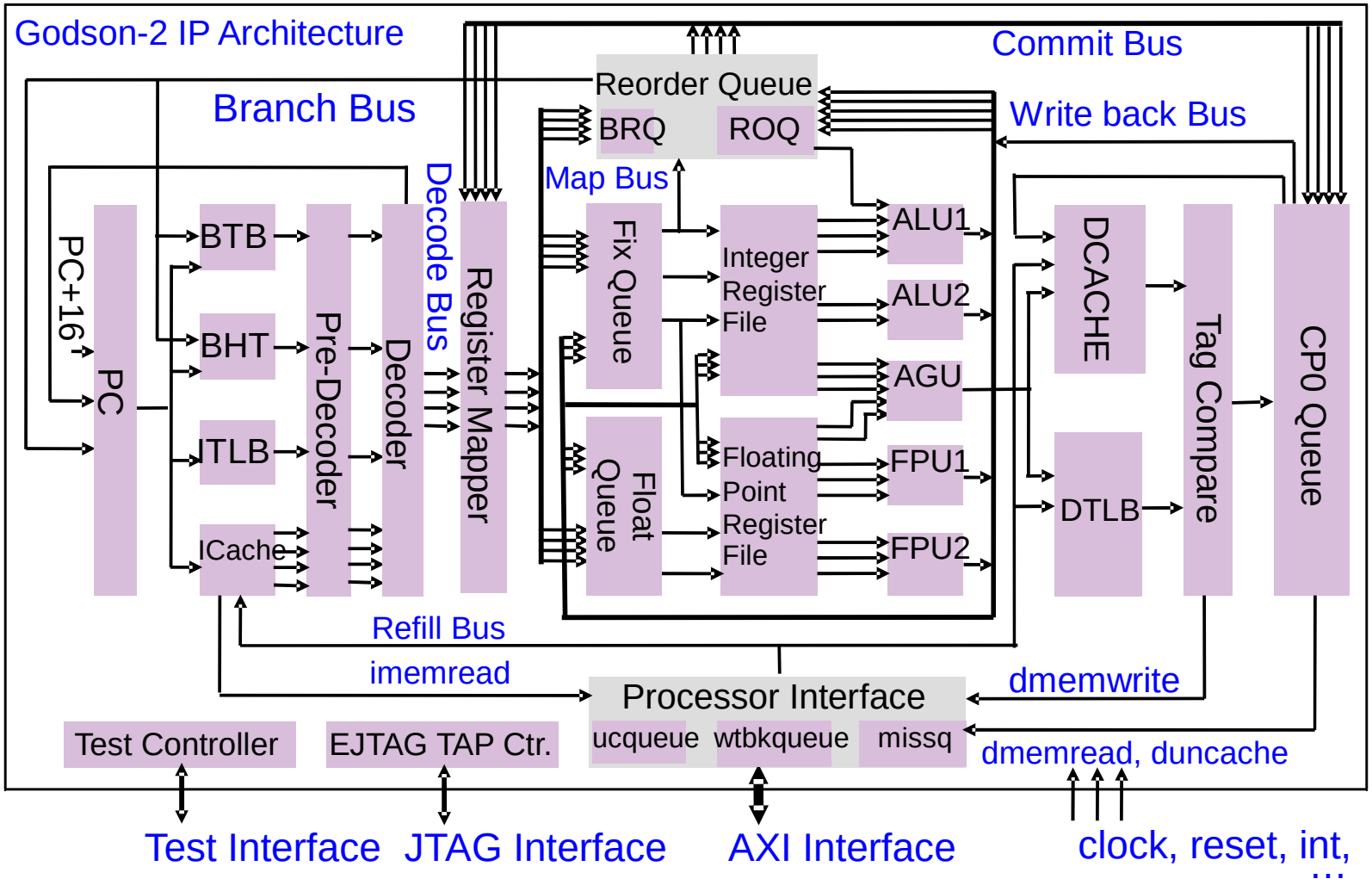
SPARC – práce s registrovými okny



Varianty na bázi MIPS

- Pravděpodobně jeden čas architektura s nevěším počtem užití na Zemi (různé AP, embedded systémy atd.)
- Stále ve vývoji i pro výkonné desktopy a superpočítače – Loongson3A
- MIPS Aptiv – MIPS32 pro vestavné aplikace
- MIPS Warrior – MIPS P6600 MIPS64 Release 6 – hardware virtualization with hardware table walk, 128-bit SIMD
- MIPS inspiroval mnoho SoftCore jader pro FPGA
  - Xilinx Microblaze
  - Altera Nios

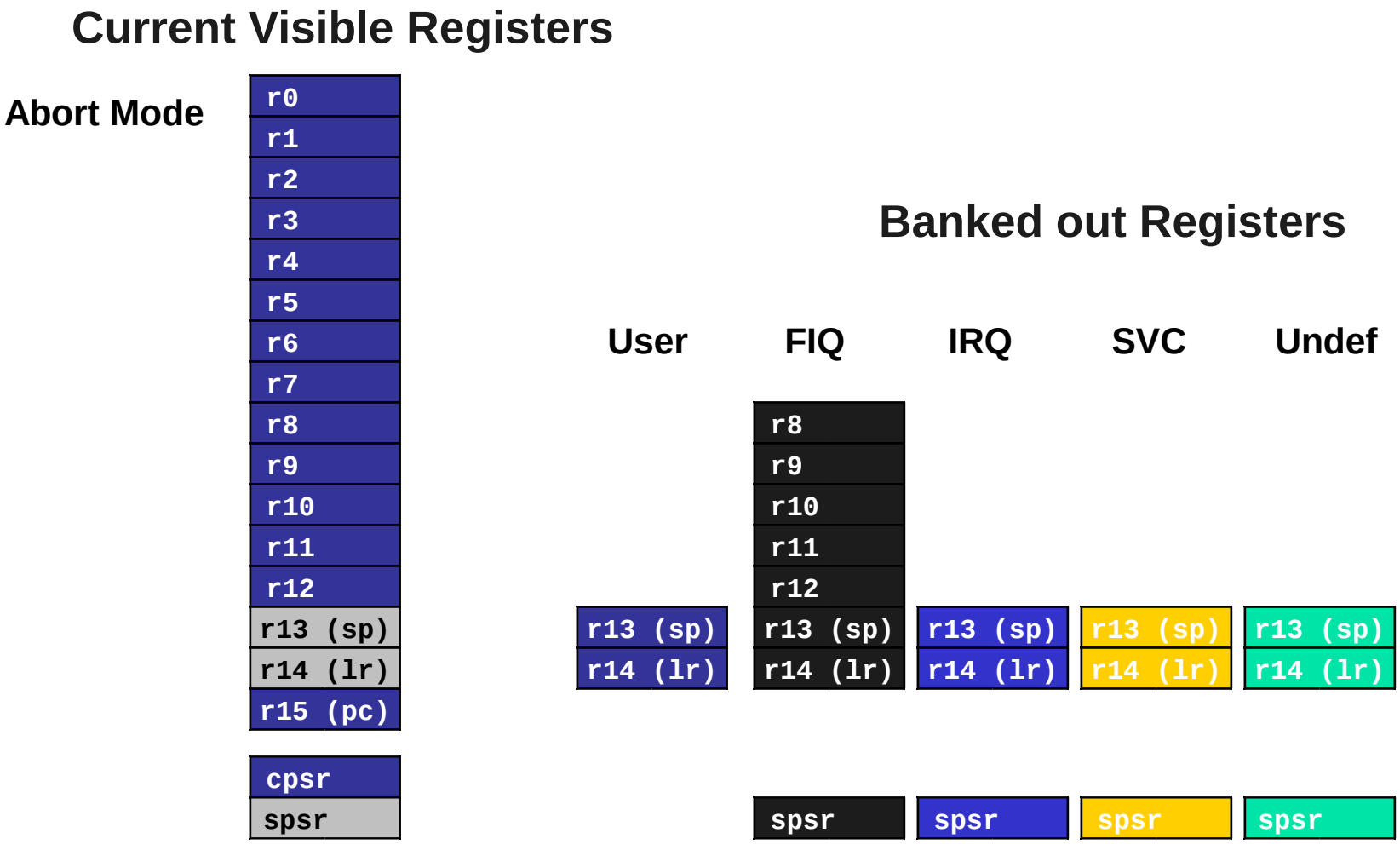
Loongson3A


Godson-2 IP Architecture
Commit Bus
Reorder Queue
BRQ
ROQ
Branch Bus
Write back Bus
Map Bus
Decode Bus
PC+16
PC
BTB
BHT
ITLB
ICache
Pre-Decoder
Decoder
Register Mapper
Fix Queue
Float Queue
Integer Register File
Floating Point Register File
ALU1
ALU2
AGU
FPU1
FPU2
DCACHE
DTLB
Tag Compare
CP0 Queue
Refill Bus
imemread
dmemwrite
Processor Interface
ucqueue
wtbkqueue
missq
dmemread, duncache
Test Controller
EJTAG TAP Ctr.
Test Interface
JTAG Interface
AXI Interface
clock, reset, int, …

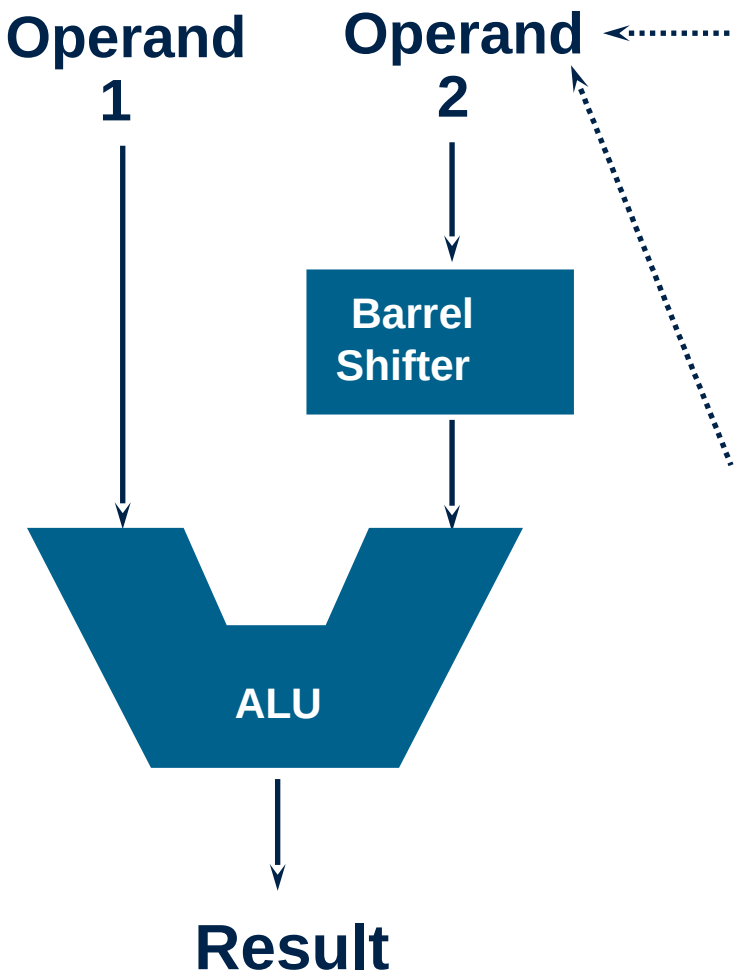
Snaha opět zkrátit instrukce ⇒ aliasy, proměnná délka u RISC, VLIW

- ARM, 16bit aliasy nejčastějších 32bit instrukcí (režim Thumb)
- M-Core, 32bit CPU s 16bit instrukcemi
- ColdFire - RISC implementace na bázi 68000, 16, 32, 48bit instrukce

Architektura ARM - registry



Architektura ARM – ALU a operandy



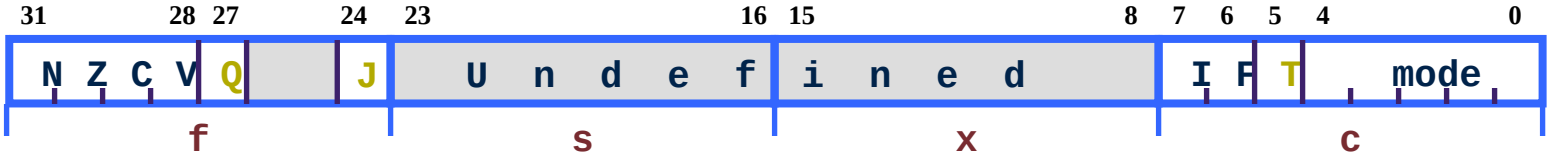
Register, optionally with shift operation

- Shift value can be either be:
  - 5 bit unsigned integer
  - Specified in bottom byte of another register.
- Used for multiplication by constant

Immediate value

- 8 bit number, with a range of 0-255.
  - Rotated right through even number of positions
- Allows increased range of 32-bit constants to be loaded directly into registers

Architektura ARM – stavové slovo



- Condition code flags
  - N = Negative result from ALU
  - Z = Zero result from ALU
  - C = ALU operation Carried out
  - V = ALU operation oVerflowed
- Sticky Overflow flag - Q flag
  - Architecture 5TE/J only
  - Indicates if saturation has occurred
- J bit
  - Architecture 5TEJ only
  - J = 1: Processor in Jazelle state
- Interrupt Disable bits.
  - I = 1: Disables the IRQ.
  - F = 1: Disables the FIQ.
- T Bit
  - Architecture xT only
  - T = 0: Processor in ARM state
  - T = 1: Processor in Thumb state
- Mode bits
  - Specify the processor mode

Architektura ARM – operační režimy

- User : unprivileged mode under which most tasks run
- FIQ : entered when a high priority (fast) interrupt is raised
- IRQ : entered when a low priority (normal) interrupt is raised
- Supervisor : entered on reset and when a Software Interrupt instruction is executed
- Abort : used to handle memory access violations
- Undef : used to handle undefined instructions
- System : privileged mode using the same registers as user mode

Téměř závěr

- Neexistuje ideální řešení všech diskutovaných problémů vhodné pro všechna nasazení
- Nezbývá než všechny výše uvedené techniky kombinovat a optimalizovat podle oblasti použití procesoru (maximální výpočetní výkon/minimální spotřeba)
- heterogenní systémy pro výpočetně náročné aplikace – vektorové jednotky, GPU, FPGA akcelerátory

ARM 64-bit – AArch64

- K volání používá LR, ruší banky registrů, pro výjimky ELR
- PC vyčleněn zvlášť (není již běžný registr)
- 31 64-bitových registrů R0 až R30 (R30 = X30 ≅ LR)
  - V kódu W*n* (W0) pro 32-bit přístup, X*n* (X0) pro 64-bit
  - Reg. code 31 stejně jako MIPS 0, v kódu WZR/XZR
  - V některých instrukcích má reg. code 31 význam WSP, SP
- Přímý operand 12-bit s volitelným LS 12 pro aritmetické operace, opakující se bitové masky pro logické
- 32-bit operace ignorují bity 32–63 na vstupu a nulují na výstupu

AArch64 – Větvení a podmíněné operace

- Zrušená možnost podmínění každé instrukce i Thumb IT
- Zůstává příznakový registr, přidané CBNZ, CBZ, TBNZ, TBZ
- Pouze několik podmíněných instrukcí
  - součet a rozdíl s přenosem, select (move C?A:B)
  - nastavení 0 a 1 (nebo -1) podle podmínky
  - podmíněná instrukce porovnání
- 32-bit a 64-bit násobení a dělení (3 registry), násobení se sčítáním $64 \times 64+64 \rightarrow 64$ (čtyři registry), bity 64 až 127 z výsledku násobení

AArch64 – Přístup do paměti

- Adresa 48+1 bit, znaménkové rozšíření na 64-bitů
- Přímo zakódovaný offset voliteně násoben šířkou přístupu
- Když je použitý registr jako index, může být násobený šířkou přístupu a také lze omezit na 32-bitů
- PC relative ±4GB lze zakódovat do 2 instrukcí
- Pouze načtení dvou nezávislých registrů LDP a STP (zrušeno LDM, STM), přidané LDNP, STNP
- Podporuje nezarovnané přístupy
- LDX/STX(RBHP) pro exkluzivní přístup 1,2,4,8 a 16 bytů

AArch64 – Adresní režimy

- Simple register (exclusive)

  [base{,#0}]

- Offset

  [base{,#imm}] – Immediate Offset

  [base,Xm{,LSL #imm}] – Register Offset

  [base,Wm,(S|U)XTW {#imm}] – Extended Register Offset

- Pre-indexed

  [base,#imm]!

- Post-indexed

  [base],#imm

- PC-relative (literal) load

  label

Bits	Sign	Scaling	WBctr	LD/ST type
0	-	-	-	LDX, STX, acquire, release
9	signed	scaled	option	reg. pair
10	signed	unscaled	option	single reg.
12	unsig.	scaled	no	single reg.

RISC-V – znovu zjednodušit/optimalizovat RISC

- Patterson, Berkeley RISC 1984 → počátek RISC éry, vyvinul se ve **SPARC** (Hennessy **MIPS**, Stanford University)
- Komercializací o rozvojem se architektury staly opět příliš složité, zároveň současné licence a patenty znemožňují původním tvůrcům vlastní reálnou implementaci ve formě křemíku pro výuku a výzkum
- MIPS je základem většiny základních kurzů a často je i implementace podobného procesoru součástí pokročilejších kurzů (A4M36PAP)
- Krste Asanovic a další studenti Dr. Pattersona pro začali vyvíjet vlastní novou architekturu (počátek 2010)
- BSD Licence, proto, aby byla přístupná i v budoucnu
- Podpora GCC, binutils., Linux, QEMU, atd.
- Proti SPAC jednodušší, více podobná MIPS ale optimalizovaná na úrovni zátěže hradel (fanout) a délky kritických cest v budoucích návrzích
- Existuje již několik otevřených implementací a projekt **lowRISC** má za cíl výzkum v oblasti bezpečnosti a implementaci SoC na křemíku

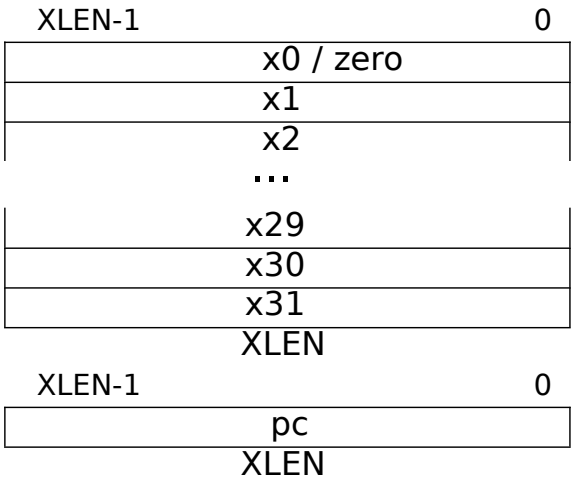
RISC-V – specifikace architektury

- Specifikace ISA dostupná na http://riscv.org/
  - The RISC-V Instruction Set Manual, Volume I: User-Level ISA, Version 2.0

    Andrew Waterman, Yunsup Lee, David Patterson, Krste Asanovic
  - Ne jen popis architektury, ale i důvodů proč bylo dané řešení vybrané a jaké problémy/cenu obnáší alternativní řešení
- klasický návrh 32 celočíselných registrů, jeden nula (zero), operandy regsrc1, regsrc2, regdest, zajímavost striktní dodržení i pro SaveWord,důsledkem nespojitost přímých/immediate operandů, PC mimo registry, PC-relativní adresování
- Varianty s 32, 64 a 128-bitovými registry a adresací
- Vysoká hustota kódu (plánovaná 16-bit varianta kódování instrukcí)
- V kódování systematicky vyhrazen prostor pro plovoucí řádovou čárku (single, double, quad) a multimediální SIMD instrukce atd.

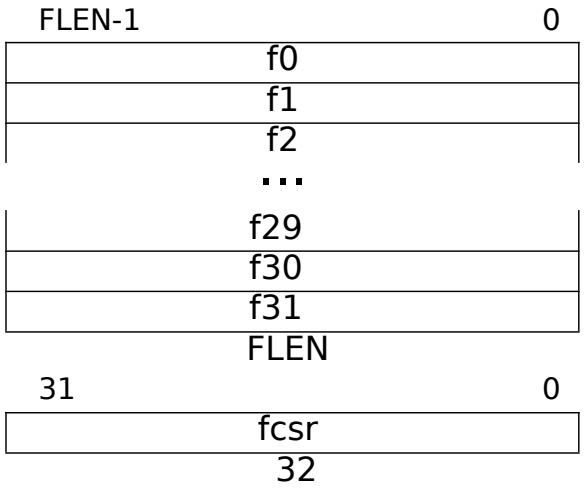
RISC-V – registers

Integer registers



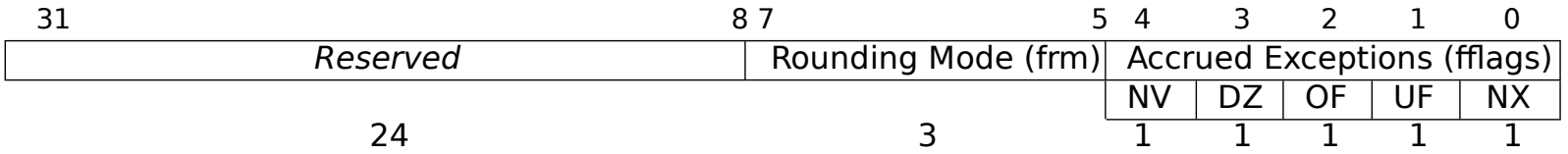
Variant	XLEN
RV32	32
RV64	64
RV128	128

Floating point registers



Variant	FLEN
F	32
D	64
Q	128

Floating-point control and status register

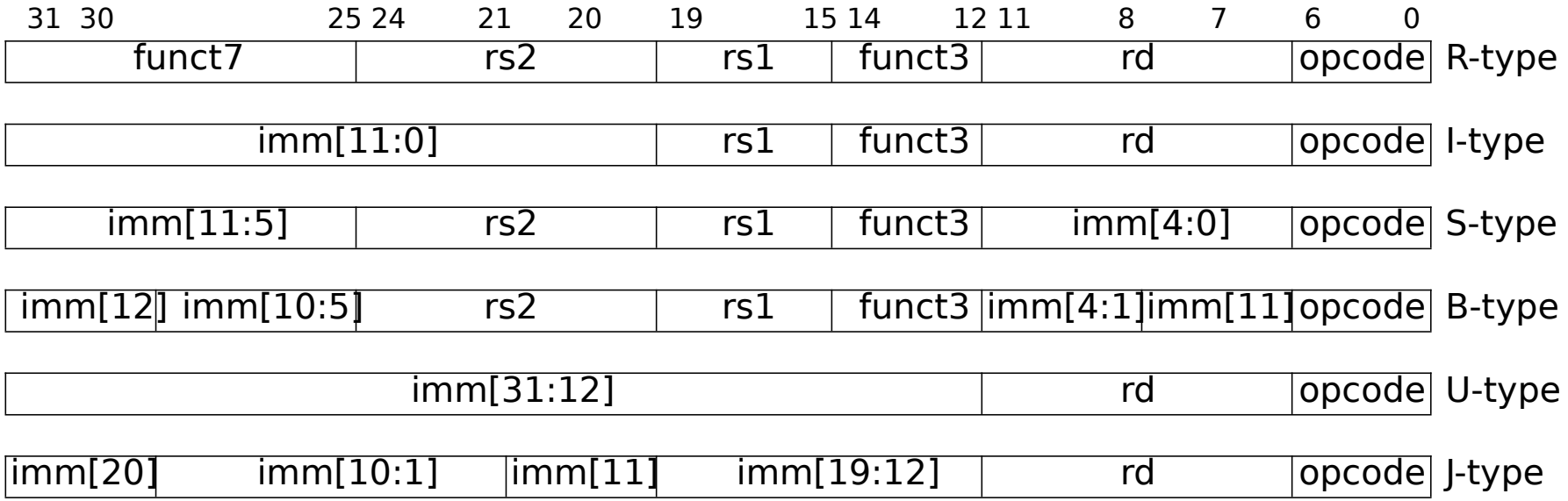


Source: https://riscv.org/specifications/

RISC-V – instruction length encoding

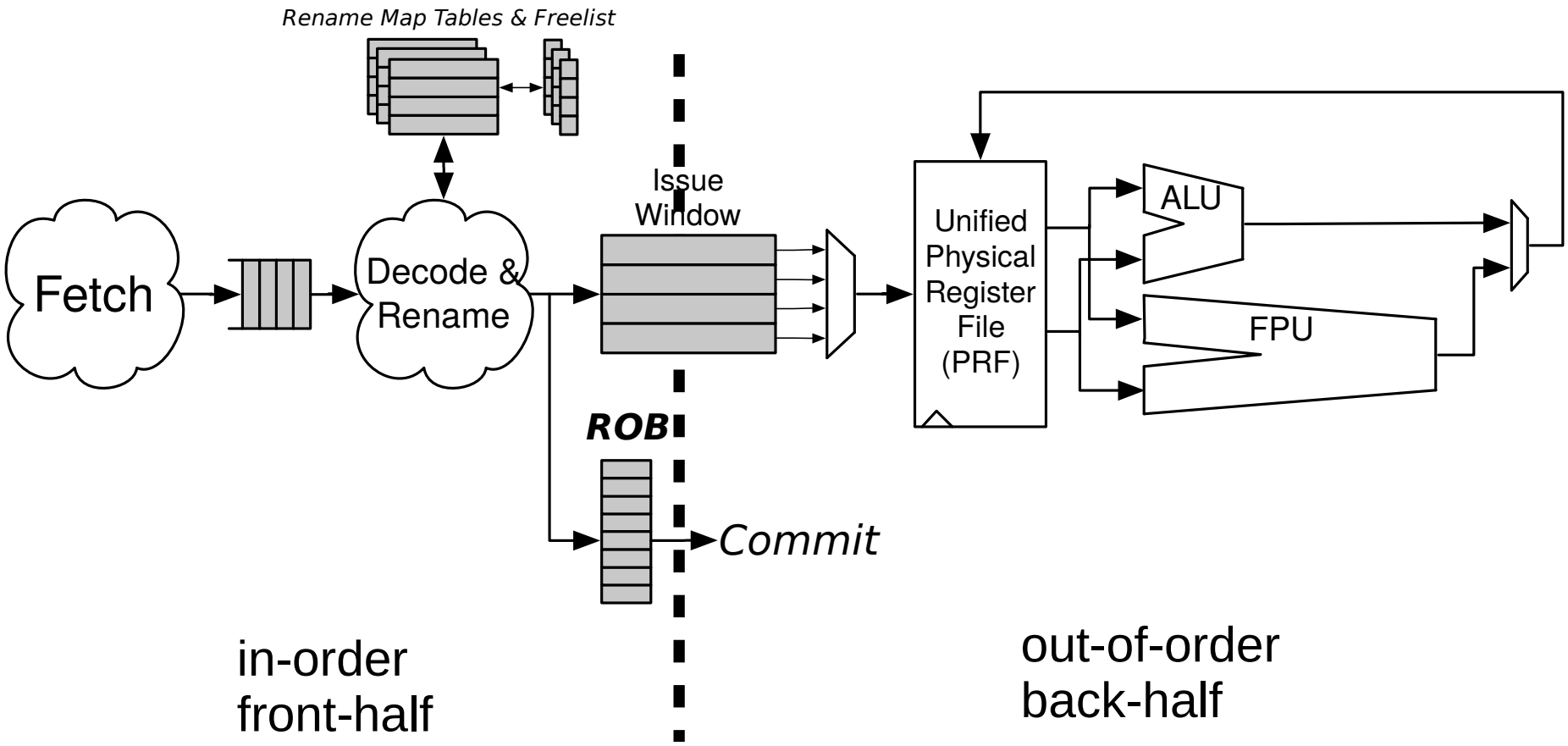
Source: https://riscv.org/specifications/

RISC-V – 32-bit instructions encoding



Source: https://riscv.org/specifications/

BOOM Superscalar RISC-V into Rocket Chip



Main developer: Christopher Celio

9k source lines + 11k from Rocket

Source: https://riscv.org/wp-content/uploads/2016/01/Wed1345-RISCV-Workshop-3-BOOM.pdf