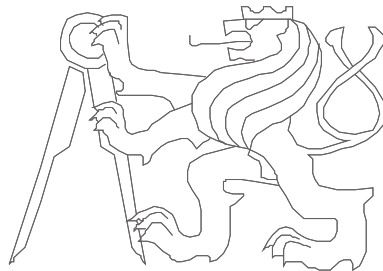


Pokročilé architektury počítačů

06

Superskalární organizace - úvod
(diverzifikované dynamické zřetězení)



České vysoké učení technické, Fakulta elektrotechnická

Skalární organizace

- Limity skalárního zřetězení:
 - propustnost je ohraničena na 1 IPC
 - unifikace – musí podporovat všechny instrukce – neefektivní..
 - pozastavení (stall) vede ke vzniku nevyužití předcházejících stupňů pipeline (tzv. rigid pipeline)

Programová propustnost:

$$W = 1 / T = \text{IPC} \cdot f_{\text{CLK}} / \text{IC}$$

- hlubší pipeline dává sice možnost zvýšit pracovní frekvenci (méně hradel v každém stupni), ale...

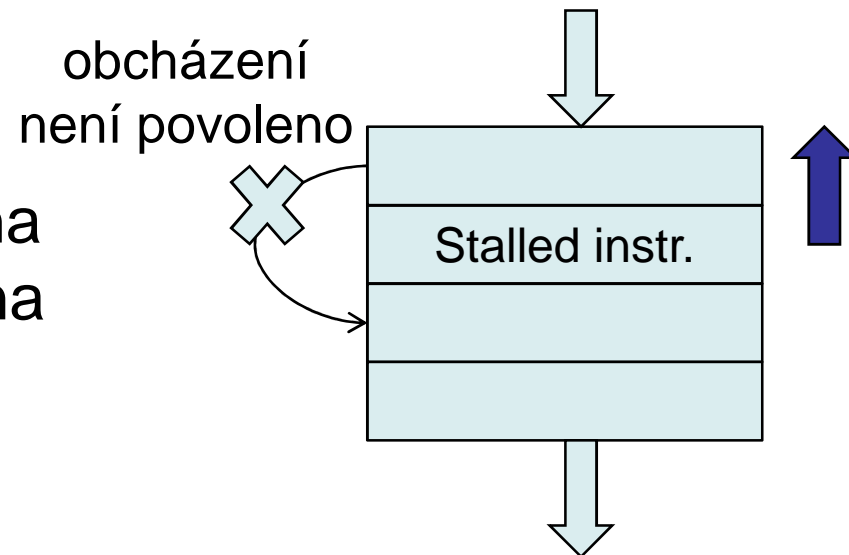
Skalární organizace

- Problém unifikace:
 - Rozdílné typy instrukcí požadují rozdílný způsob zpracování. Je obtížné a zároveň neefektivní tyto různé požadavky implementovat v unifikované pipeline.
 - Rozdílnost latencí: Určité typy instrukcí požadují vícero vykonávacích cyklů (instrukce v pohyblivé řádové čárce a určité instrukce v pevné řádové čárce – násobení, dělení), zatímco jiné mají latenci jenom jednoho cyklu...
 - Přirozeným řešením je vytvořit paralelní pipeline, ne však unifikované, ale **diverzifikované** (každá paralelní větev zřetězení je specializována na vykonávání jiné funkce).

Skalární organizace

- Problém vzniku pozastavení (stall) – nepřizpůsobivá pipeline
 - skalární pipeline jsou „tuhé“ v smyslu, že instrukce procházejí přes stupně pipeline jedna za druhou a vznik pozastavení v určitém stupni se „šíří“ do předchozích stupňů

div **R0**, R1, R2
sub R3, **R0**, R4 - blokována
add R5, R6, R7 - blokována



- instrukce jsou vykonány v programovém pořadí...

Superskalární organizace

Superskalární zřetězení:

- vícero funkčních jednotek řazeno paralelně (umožňuje simultánně zpracovávat vícero instrukcí napříč stupni zřetězení)
- významným atributem je schopnost vykonávat instrukce v jiném pořadí než určuje původní program – tzv. **dynamické zřetězení**
- schopnost inicializovat vykonávání vícero instrukcí v jednom cyklu
- představuje kombinaci časového a prostorového paralelizmu
- **šířka** zřetězení je rovna počtu paralelních větví (počtu instrukcí, které mohou být přineseny, dekodovány nebo dokončeny v každém cyklu)

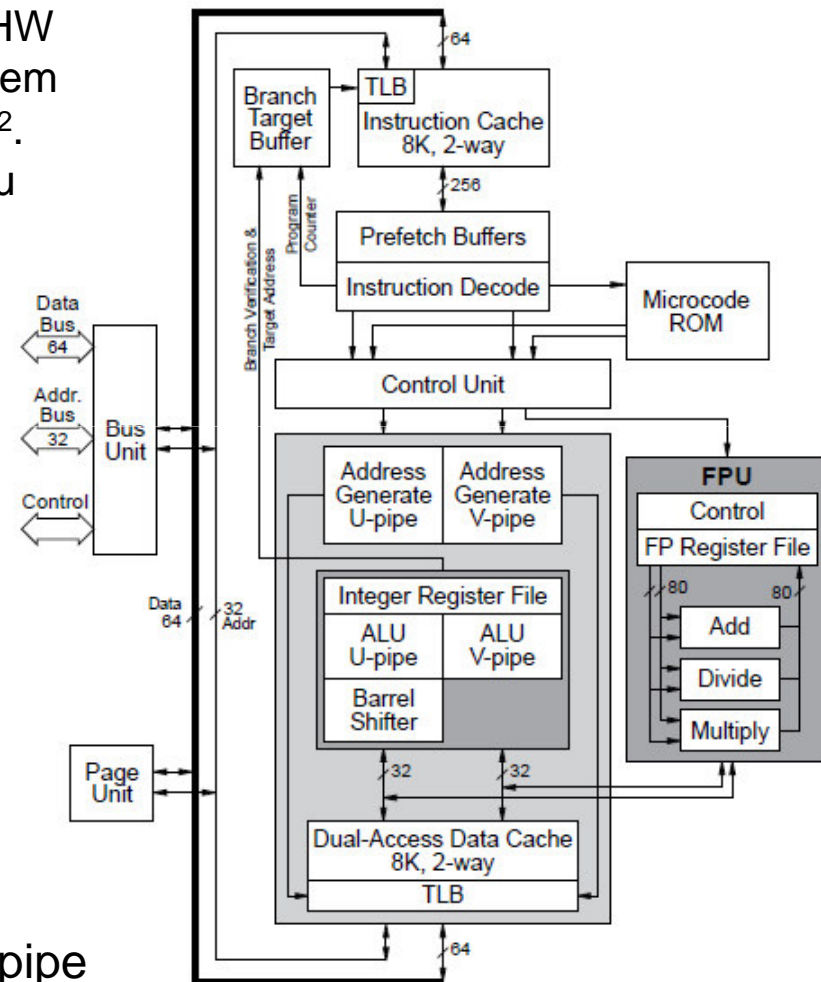
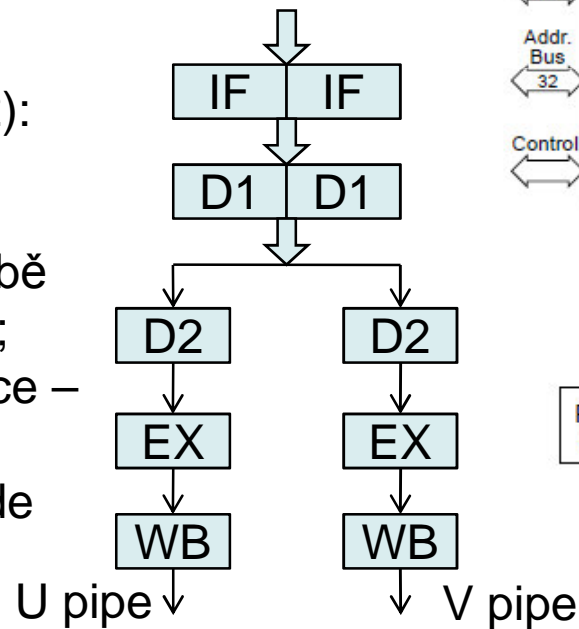
Superskalární organizace

- Vyžaduje dodatečné HW prostředky pro implementaci.

(Pokud označíme šířku zřetězení w , potom cena HW v každém stupni může potenciálně narůstat faktorem w , a cena mezistupňového propojení faktorem w^2 . Taktéž počet čtecích a zapisovacích portů souboru pracovních registrů musí být navýšen faktorem w , podobně přístupové porty I-cache a D-cache.)

5-stupňové paralelní zřetězení Pentia ($w=2$):
(předchůdce: i486)

Dvě ALU operace – obě čtou pracovní registry;
Dvě paměťové operace – duální přístup.. Ale?
(serializace – v případě konfliktů bank)

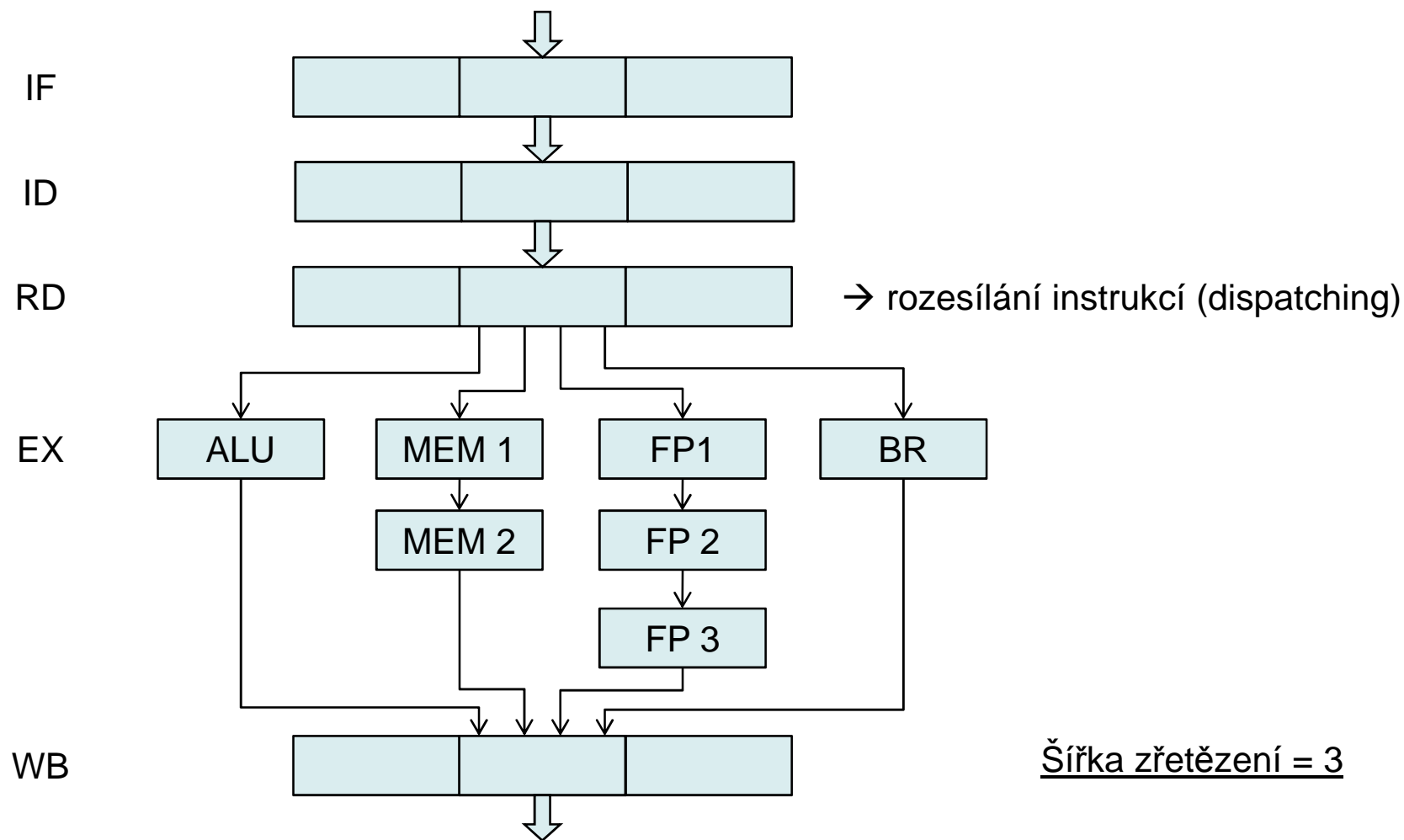


Superskalární organizace – **diverzifikované zřetězení**

- Vykonávání určité instrukce vyžaduje pouze některé HW prostředky z jednotlivých stupňů unifikované pipeline
- Místo vytváření w totožných paralelních zřetězení (zřetězení šířky w) je vytvořena diverzifikovaná pipeline šířky w .
- Toto vede k:
 - efektivnějšímu využití HW prostředků,
 - rozdílnosti latencí jednotlivých větví („rychlejší“ instrukce mají menší latenci),
 - odstranění vzniku pozastavení (stall) nezávislých instrukcí jiných větví (nezávislost paralelních větví) – mezi-instrukční závislosti musí být řešeny před rozdělením do paralelních větví,
 - nezávislému a distribuovanému řízení.

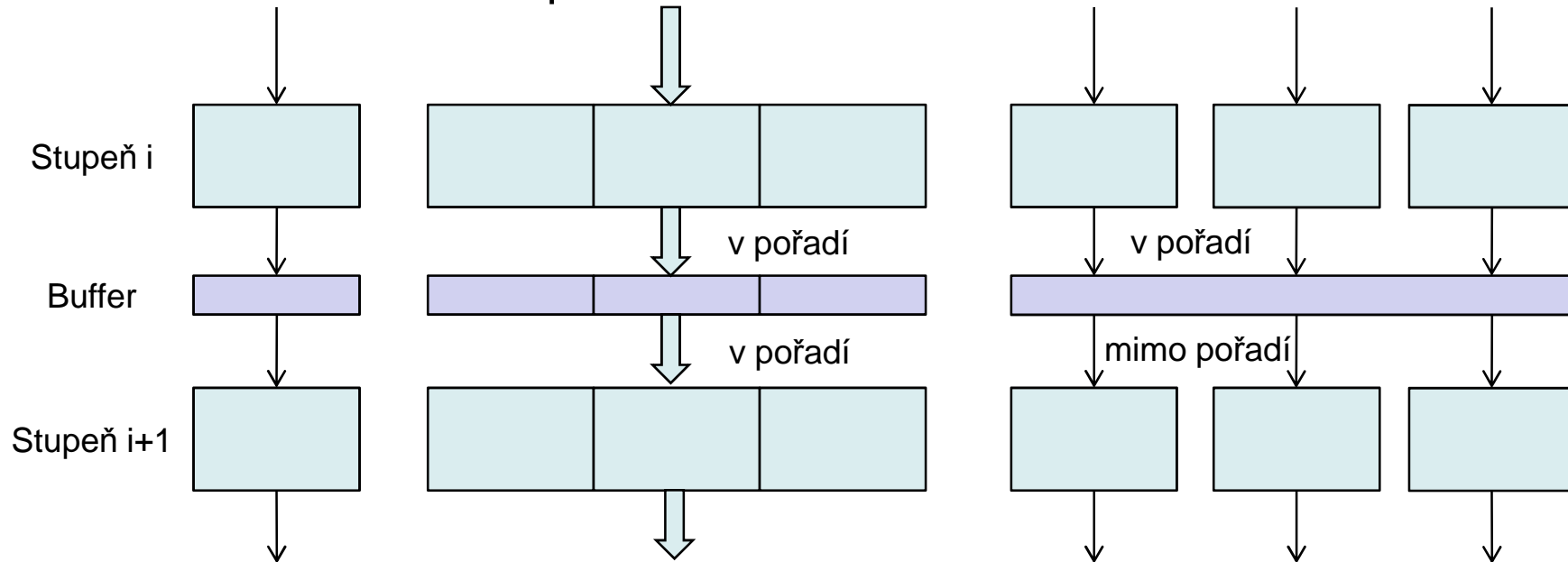
Superskalární organizace – diverzifikované zřetězení

Diverzifikované zřetězení se čtyřmi vykonávacími větvemi v stupni EX:



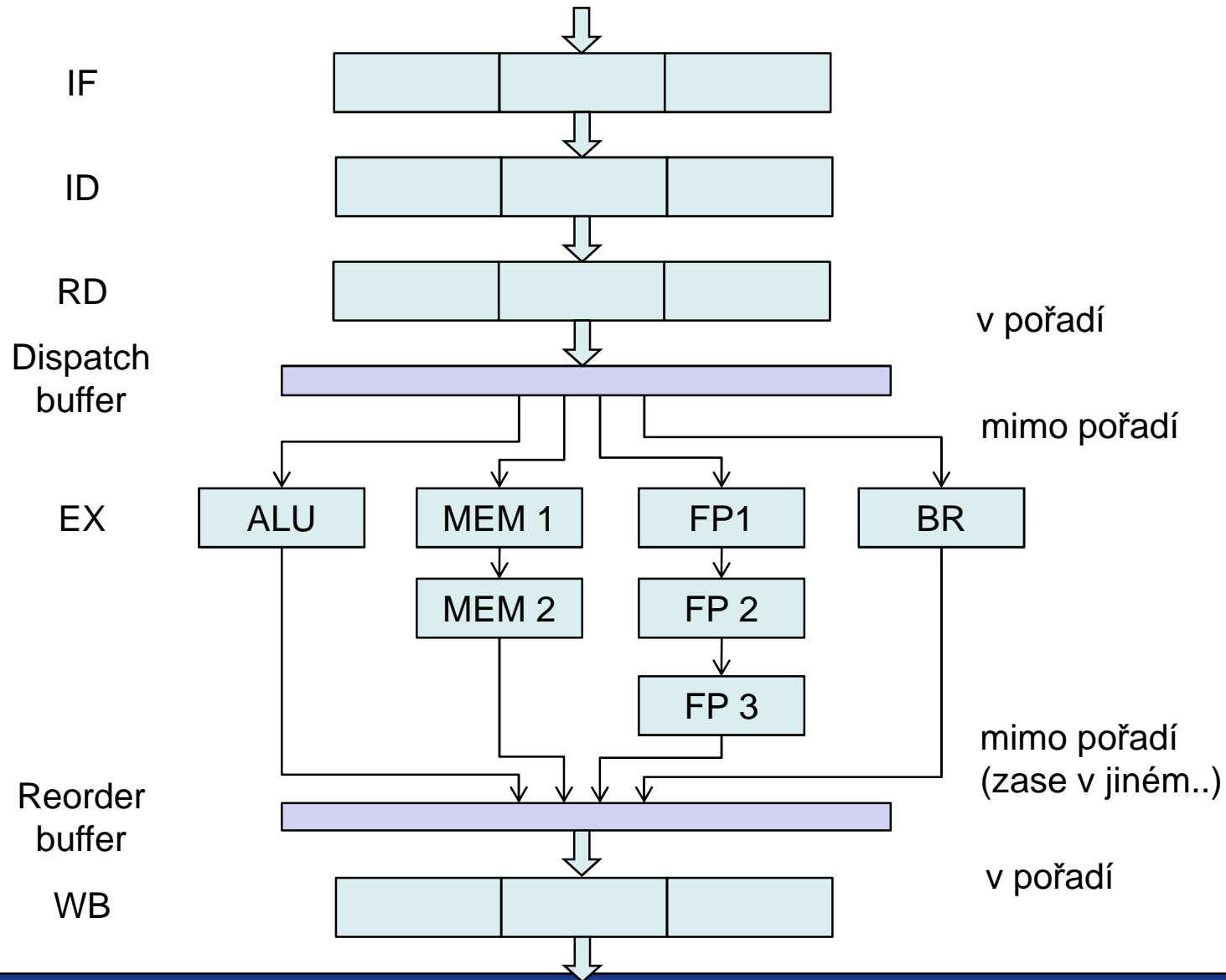
Superskalární organizace – dynamické zřetězení

- každé zřetězení (vyjma asynchronního) požaduje mezistupňové buffre mezi sousedícími stupni zřetězení...



- více-vstupový buffer: přeupořádání, nezávislý přístup (adresace), nezávislé řízení čtení a zápisu (umožňuje pozdržet instrukce..)
- dynamické zřetězení** dosahuje **vykonávání mimo pořadí** právě pomocí komplexních bufferů, které umožňují instrukcím vstupovat a vystupovat z buffru v jiném pořadí

Superskalární organizace – dynamické zřetězení

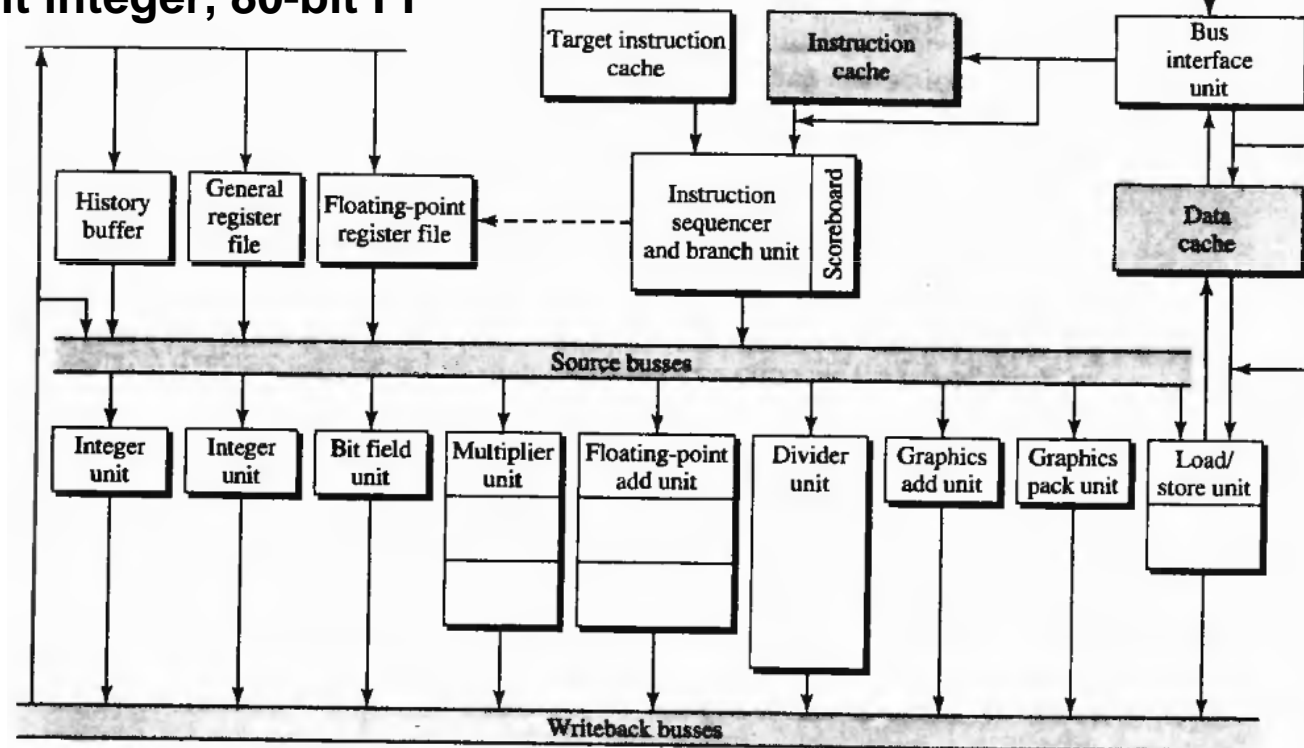


Superskalární organizace – dynamické zřetězení

- Rozesílající buffer (dispatch buffer) – instrukce přicházejí v programovém pořadí, následně jsou rozesílány jednotlivým funkčním jednotkám potenciálně v jiném pořadí..
- Funkční jednotky (diverzifikovaná pipeline) mohou mít různé latence..
- Dokončovací buffer (completion buffer, reorder buffer) slouží k zabezpečení dokončení instrukcí v programovém pořadí – nezbytné pro podporu přesného přerušení (precise exception)

Superskalární organizace

- 2 integer ALUs (32-bit operands)
- 1 FP-add (80-bit operands)
- 1 multiply (64-bit integer, 80-bit FP operands)
- 1 divide (64-bit integer, 80-bit FP operands)
- 1 bit-field (32-bit operands)
- 1 instruction/branch
- 1 data-cache
- 2 graphics (64-bit operands)

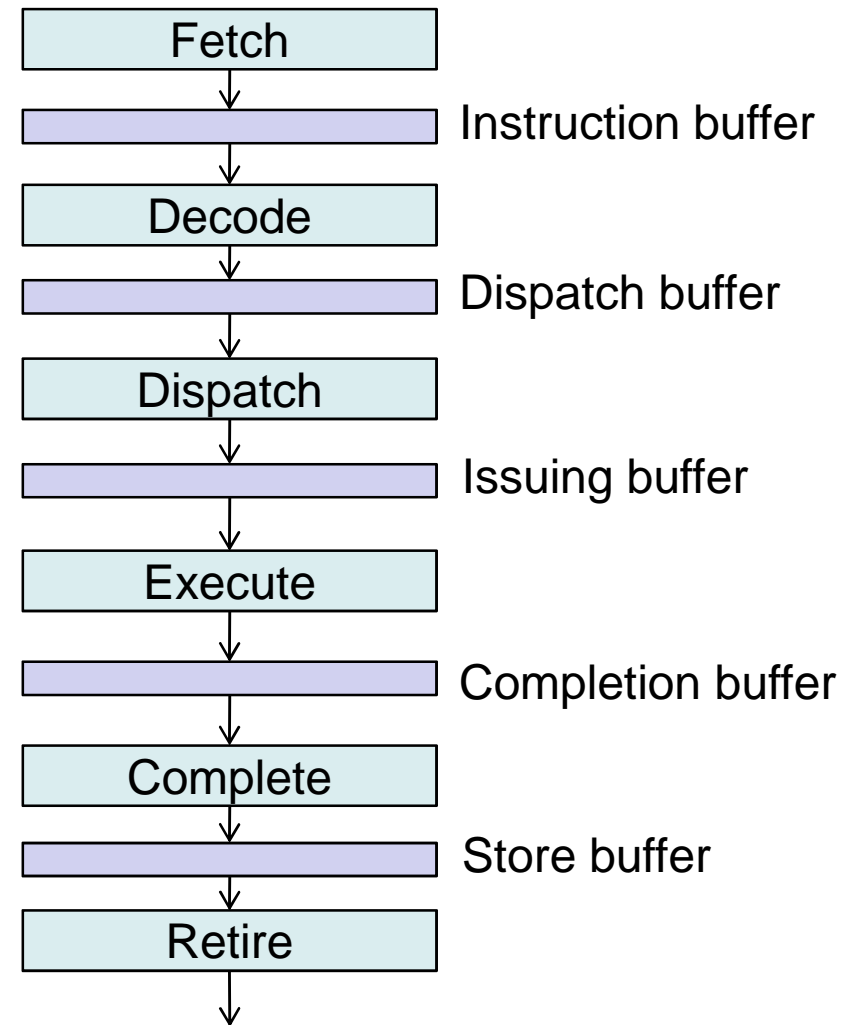


Motorola 88110,
rok 1992.

Shen, J.P., Lipasti, M.H.: Modern Processor Design : Fundamentals of Superscalar Processors, First Edition, New York, McGraw-Hill Inc., 2004

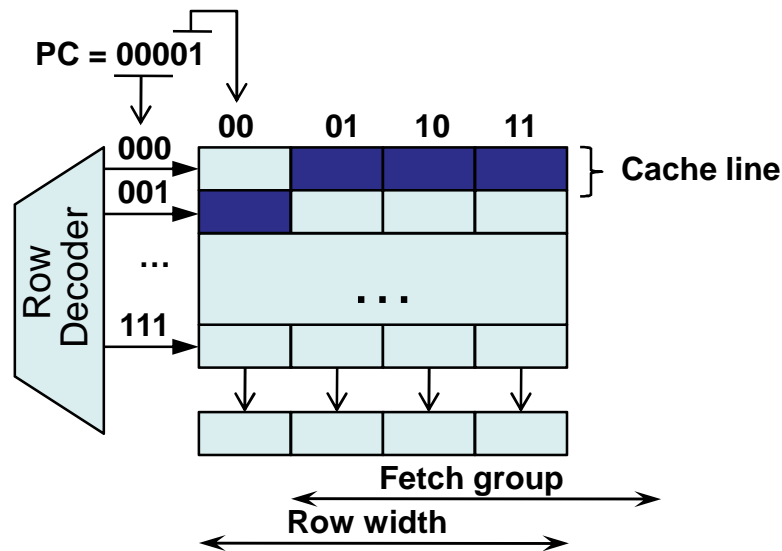
Superskalární organizace – problémy, které přináší

- Jednotlivé fáze:
 - výběr (fetch)
 - dekódování (decode)
 - rozesílání (dispatch)
 - vykonání (execute)
 - dokončení (complete)
 - „odbavení“ (retire)



Superskalární organizace – Fetch

- Superskalární zřetězení šířky w by mělo být schopné výběru w instrukcí v jednom cyklu...
- V každém cyklu stupeň Fetch používá čítač instrukcí (PC) k výběru w instrukcí - (v takzvané Fetch group)
- Propustnost všech ostatních stupňů nikdy nepřesáhne propustnost stupně Fetch – jeho propustnost snižuje nezarovnání instrukcí v I-cache a instrukce skoku (mění PC) v grupě



Možné řešení:

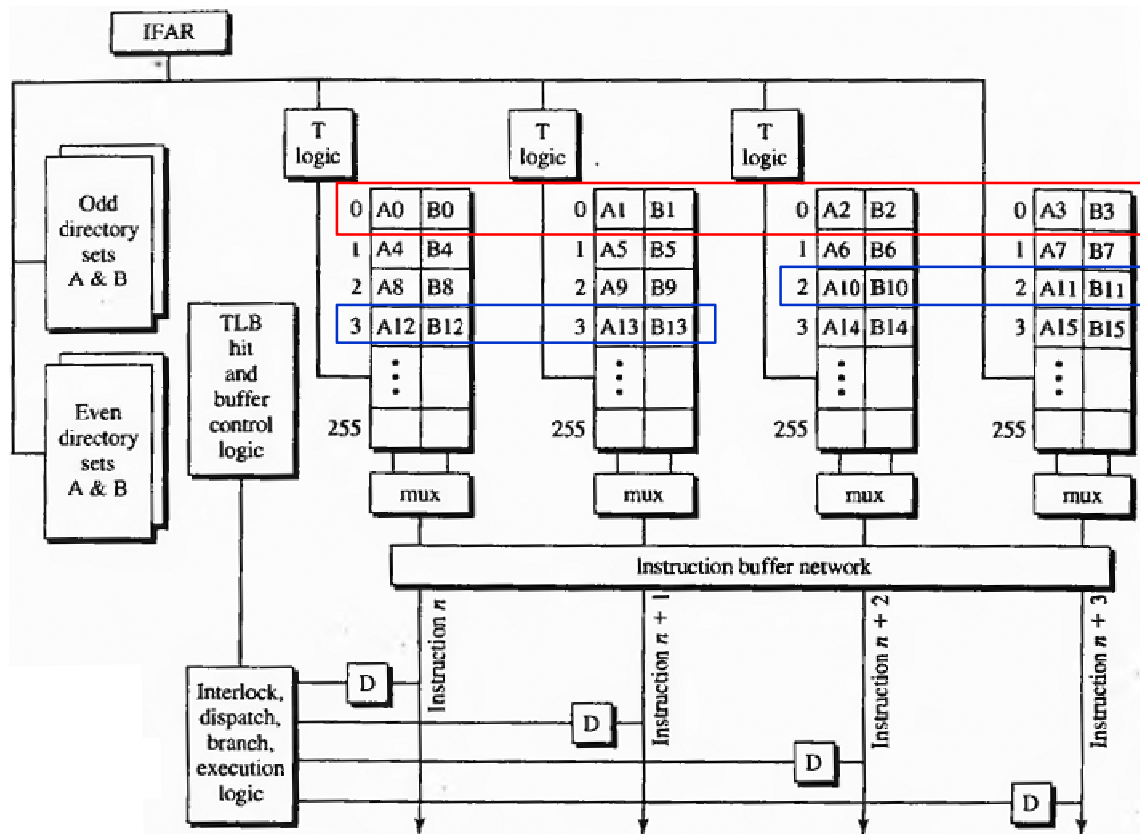
- staticky – při kompilaci – kompilátor obdrží informace o organizaci I-cache (vhodné umístění instrukcí...)
- dynamicky – hardwérem – vhodnou vnitřní organizací cache...

Superskalární organizace – Fetch

Dynamické řešení: Pokud fetch group přechází přes row boudary (ne line boundary) pak je stále možné přinést fetch group v jednom cyklu...

IBM RS/6000 (two-way set associative I-cache with Auto-Realignment)

Program:
instrukce A0
instrukce A1
instrukce A2
instrukce A3
instrukce A4
instrukce A5
...



Pokud PC adresuje A0, pak mohou být instrukce A0, A1, A2 a A3 přineseny současně bez problémů (leží na stejném řádku)

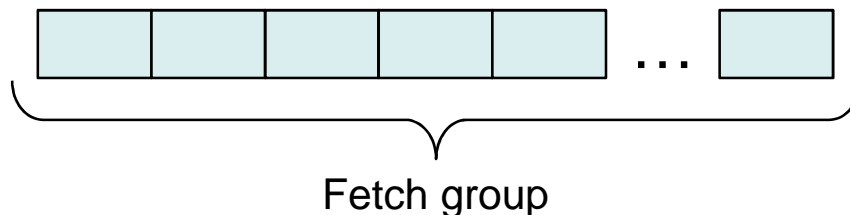
Pokud PC adresuje A10, pak instrukce A10 a A11 leží na jednom cache row, A12 a A13 na následujícím. Úkolem „T-logiky“ je detekovat tento stav a inkrementovat row index u těchto instrukcí...

Výsledkem je, že bez ohledu na adresu začátku fetch grupy budou všechny instrukce přineseny v jednom taktu (pokud nepřechází přes line boudary – pak může být nutné se obrátit na paměť). Takže pro cache line = 16 instrukcí, je 16 možných začátků fetch grupy, takže: $(13 \cdot 4 + 1 \cdot 3 + 1 \cdot 2 + 1) / 16 = 3,63$ instr/cyklus, nebo 1,188 cyklu/grupu místo $(4 \cdot 4 + 4 \cdot 3 + 4 \cdot 2 + 4) / 16 = 2,5$ instr/cyklus (nebo 1,75 cyklu/fetch grupu)

Line size: 16 instrukcí (64B), Fetch group 4 instrukce

Superskalární organizace – Decoding

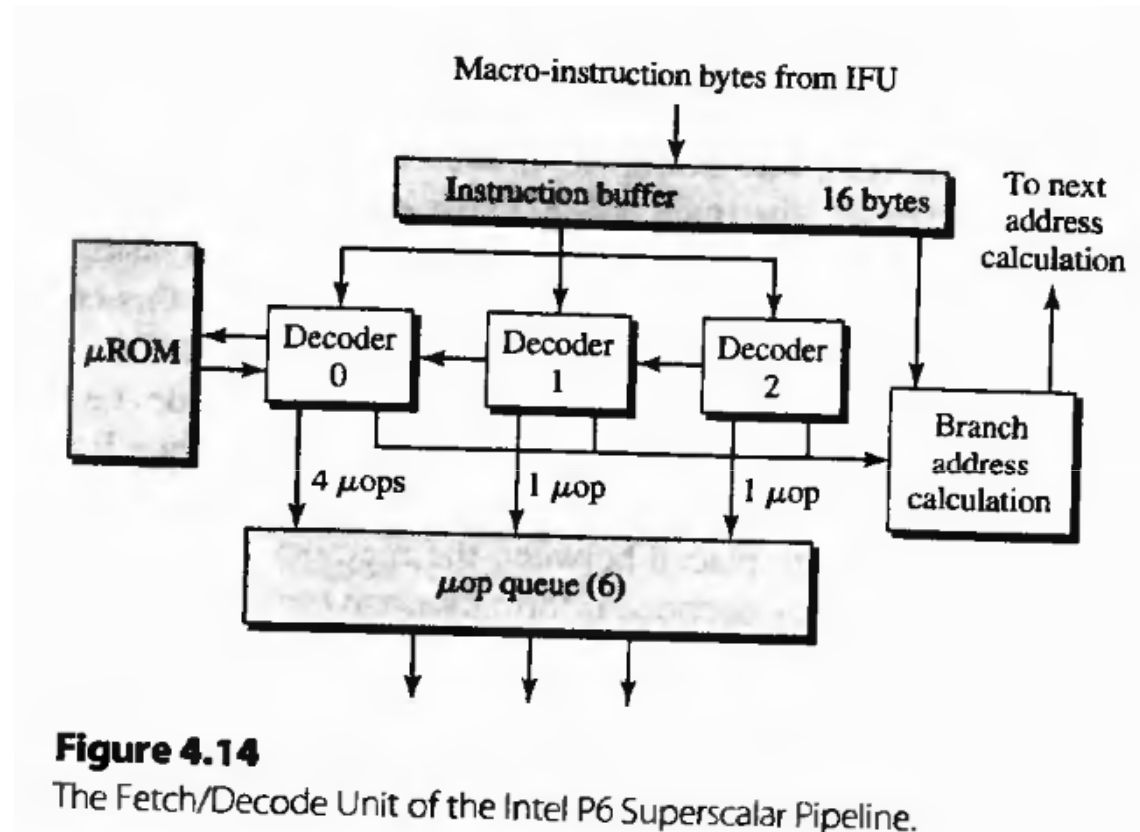
- Náročnost procesu dekódování je zejména určena: šířkou zřetězení a souborem instrukcí (ISA) - RISC vs. CISC...
- Dekódovací stupeň musí identifikovat mezi-instrukční závislosti a stanovit množinu nezávislých instrukcí, které mohou být odeslány do dalšího stupně (dispatching)
- Skupina vybraných instrukcí (fetch group) – centralizované dekódování..



CISC – identifikace instrukcí
RISC – mezi-instrukční závislosti

- vyžaduje značné množství komparátorů
- RF (Register File) musí podporovat množství simultánních přístupů
- množství sběrnic pro směrování operandů

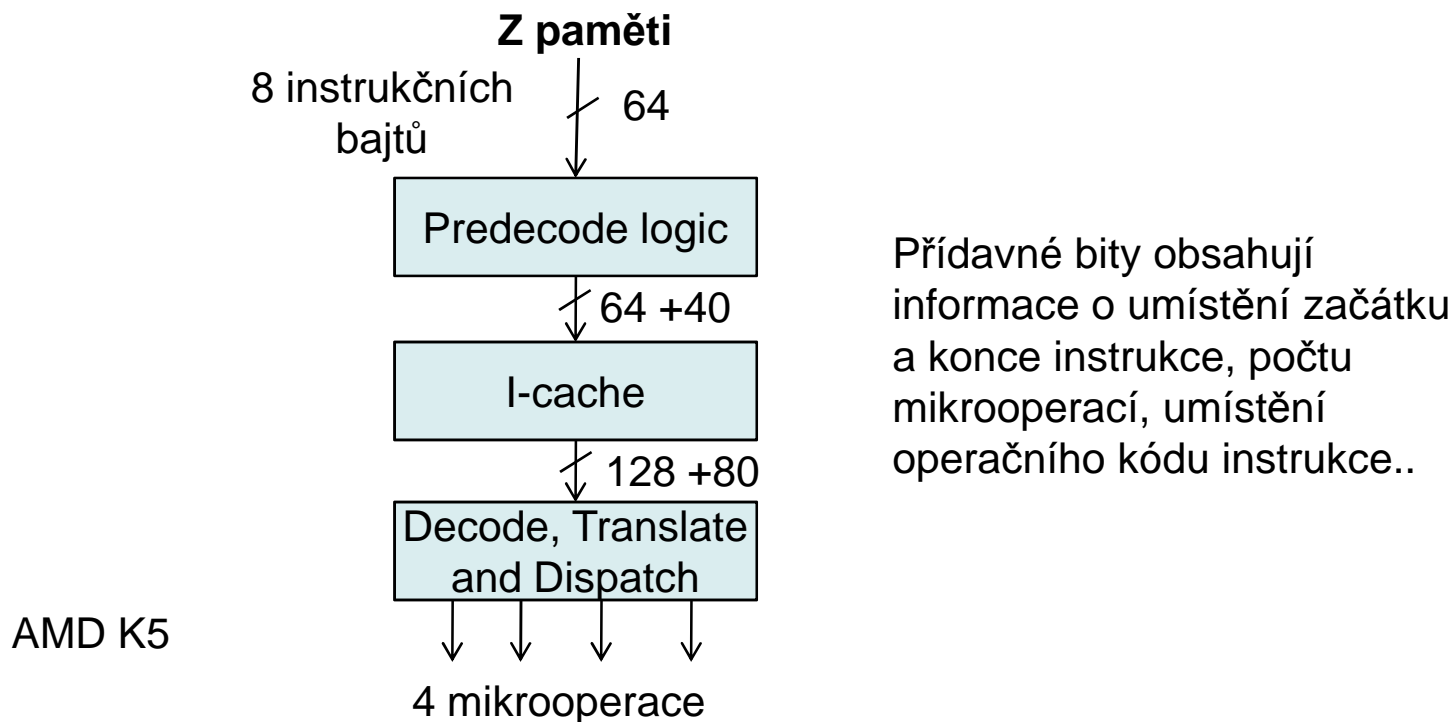
Superskalární organizace – Decoding



Shen, J.P., Lipasti, M.H.: Modern Processor Design : Fundamentals of Superscalar Processors, First Edition, New York, McGraw-Hill Inc., 2004

Superskalární organizace – Decoding

- Předdekódování instrukcí (predecoding)
 - Při vzniku I-cache miss – kdy nový řádek cache (cache line) je přenášen z paměti do I-cache, jsou instrukce částečně dekódovány
 - Uplatňuje se jak pro CISC, tak pro RISC (PowerPC, MIPS R10000) – identifikace skokových instrukcí, nezávislých instrukcí,...

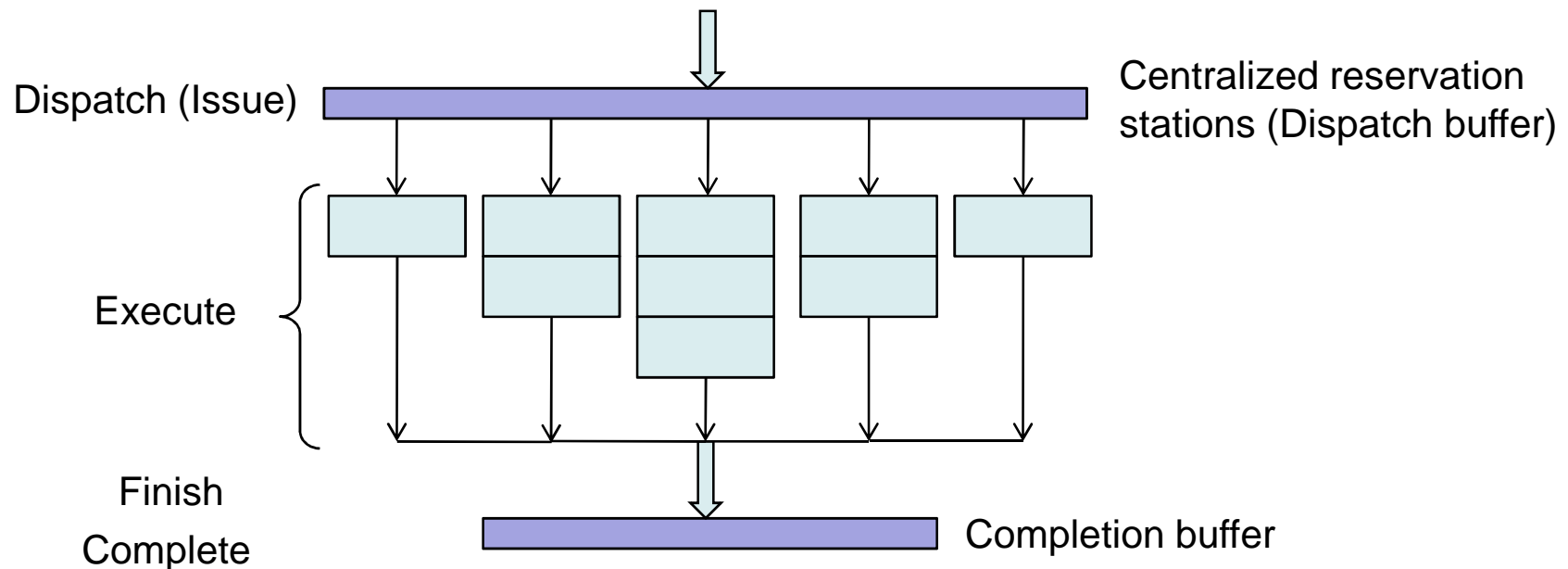


Superskalární organizace – Dispatching

- Instrukce musejí být dále směřovány svým funkčním jednotkám pro vykonání (diverzifikace) – toto je primárním cílem dispatching-u
- Některé instrukce však nemusejí mít připraveny své operandy – stall vs. dočasné buffrování (rezervační stanice) instrukcí a vykonání těch, co své operandy připraveny mají.. -> princip Data Flow architektury na nejnižší úrovni – Tomasulův algoritmus a přejmenovávání registrů (za účelem rušení WAW hazardů)
- Rozlišujeme:
 - centralizované rezervační stanice (centralized reservation station)
 - distribuované rezervační stanice (distributed)
 - hybridní (nebo též klastrované) rezervační stanice (clustered)
- Zpřesnění pojmů:
 - dispatching – asociace instrukce k funkční jednotce pro vykonání
 - issuing – inicializace vykonávání ve funkční jednotce

Superskalární organizace – Dispatching

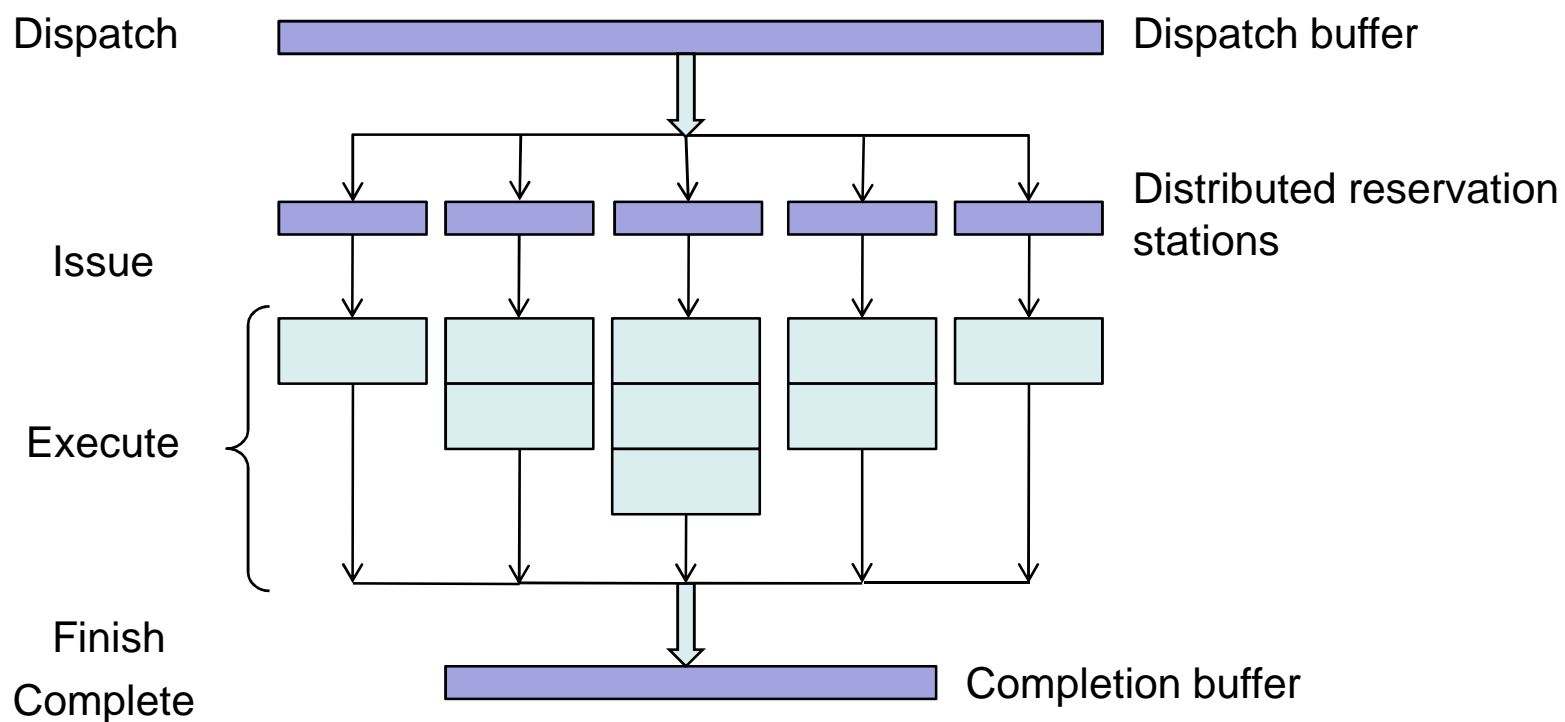
Centralizovaná rezervační stanice:



Lepší celkové využití, hardwarově náročnější

Superskalární organizace – Dispatching

Distribuovaná rezervační stanice:

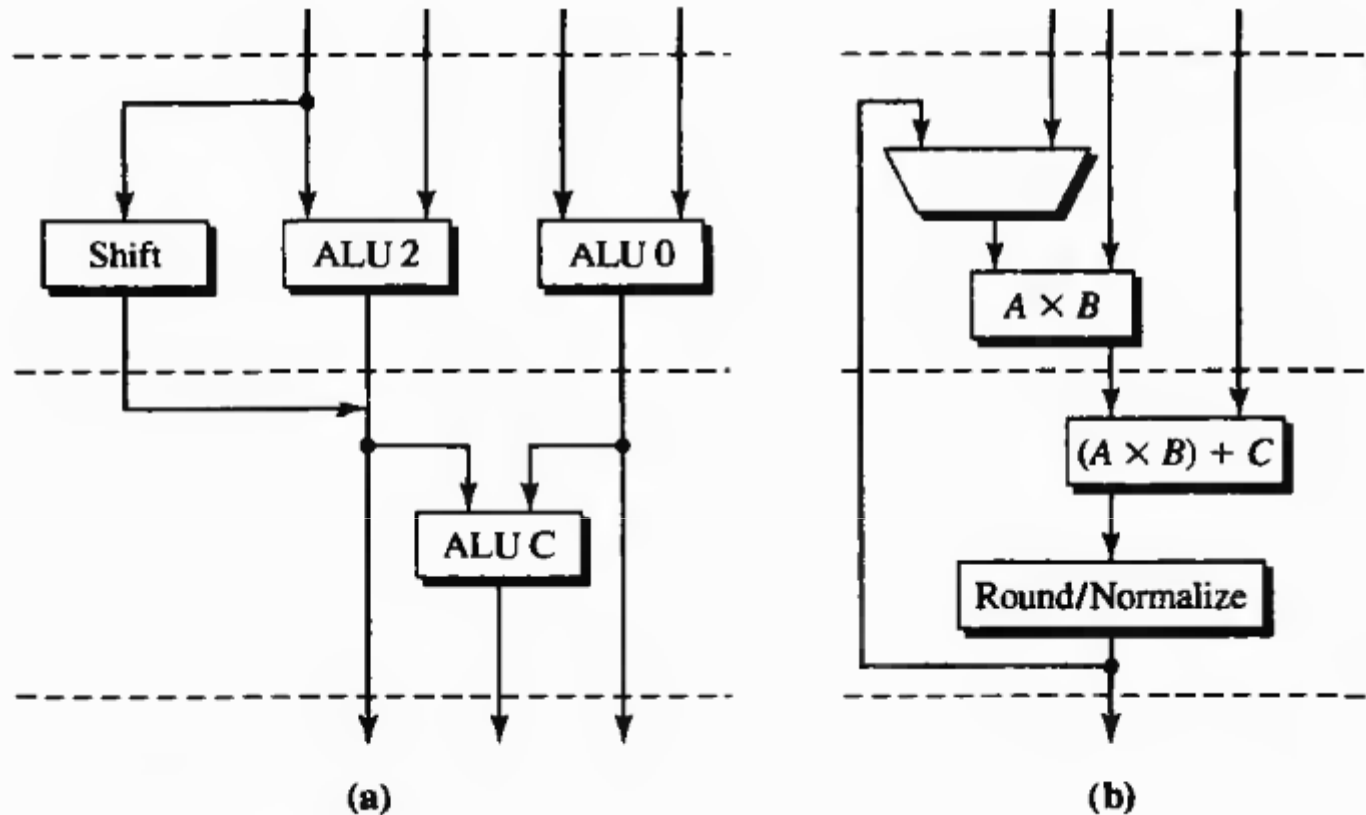


Rezervační stanice mohou být jedno-portové buffre

Superskalární organizace – Execution

- Současné procesory – více paralelních větví zřetězení, více diverzifikované (první generace – jedna větev pro celočíselné operace a jedna pro v pohyblivé řádové čárce)
- jedna větev – více než jednu operaci na více než dvou operandech..
- počet funkčních jednotek (stupeň EX) běžně převyšuje šířku pipeline
- s narůstajícím počtem funkčních jednotek narůstají dodatečné HW prostředky
 - důsledek přeposílání (forwarding) z výstupu funkčních jednotek na jejich vstup, nárůst sběrnic (mechanismus směřování operandů přináší další formu strukturálních hazardů), rezervační stanice monitorují přítomnost/dostupnost operandů (tag matching)

Superskalární organizace – Execution



(a) Integer Functional Unit in the TI SuperSPARC;

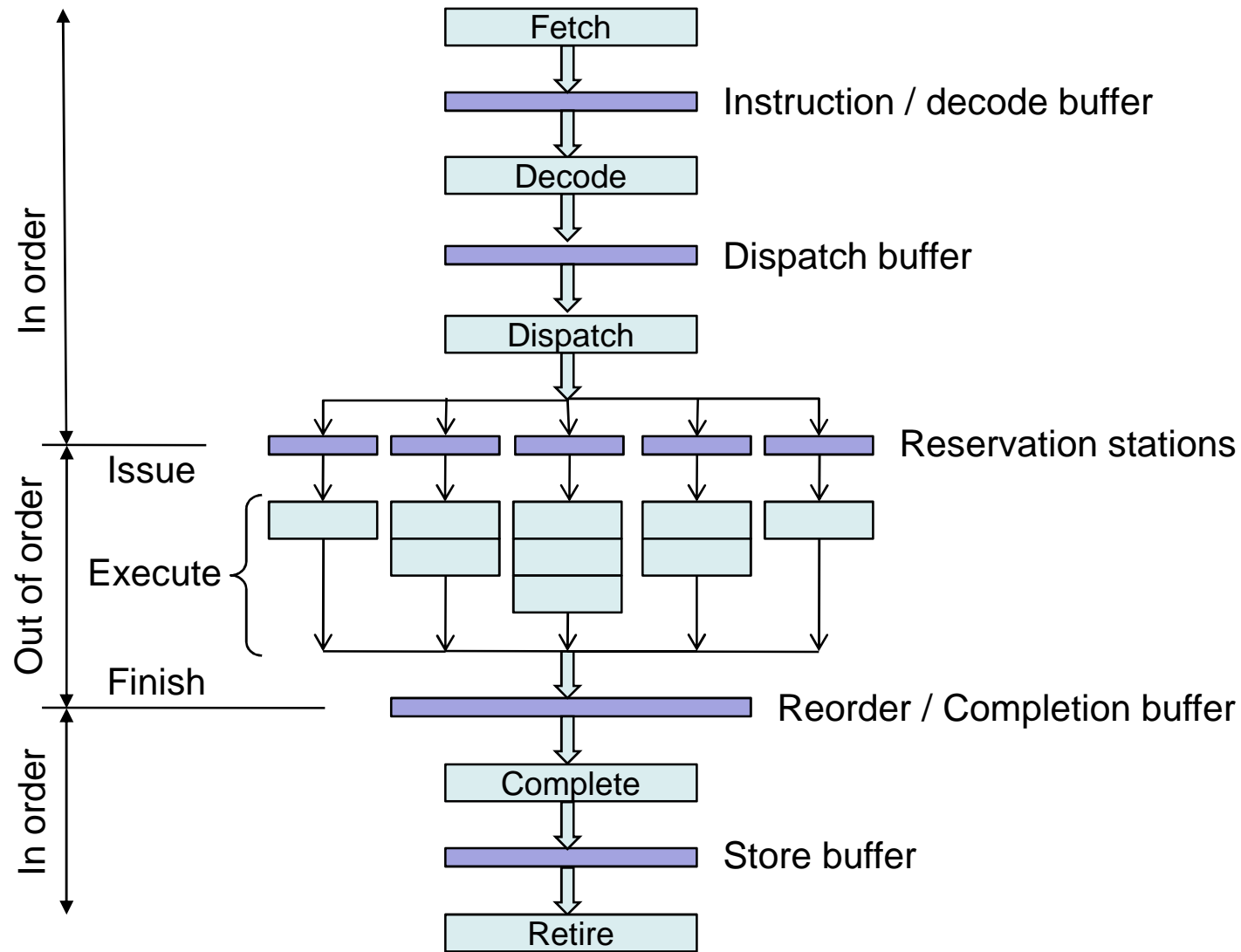
(b) Floating-Point Unit in the IBM RS/6000.

Shen, J.P., Lipasti, M.H.: Modern Processor Design : Fundamentals of Superscalar Processors, First Edition, New York, McGraw-Hill Inc., 2004

Superskalární organizace – Completion and Retiring

- Instrukce je považována za dokončenou (completed), pokud ukončí vykonávání a aktualizuje stav procesoru. (Instrukce ukončí vykonávání, pokud opustí EX stupeň a vstoupí do dokončovacího buffru.)
- Může však uplynout nějaká doba, než se výsledek dokončené instrukce projeví v D-cache. Instrukce je považována za „odbavenou“ (Retired), pokud aktualizuje D-cache.
- Tyto pojmy jsou zaměnitelné... (stejně tak dispatching a issuing, zřídka jsou pojmy Completion a Retiring posunuty o úroveň výše, tj. na Finishing a Completion)
- Problém přesného přerušení – reorder buffer.
Instrukce, která spustí přerušení je označena (tagged) v reorder buffru. Dokončovací stupeň kontroluje každou instrukci před jejím dokončením. Označené instrukci není umožněno dokončení, stav procesoru je uložen, rozpracované instrukce v pipeline jsou zahozeny.

Superskalární organizace – Completion and Retiring



Superskalární organizace – sumarizace

- Out-of-order superskalární pipeline vykonává instrukce mimo pořadí pouze mezi stupni Dispatch a Complete. Instrukce zůstávají v rezervačních stanicích po dobu jednoho nebo více cyklů čekající na své operandy. Pokud jsou tyto operandy dostupné, instrukce jsou vydány (issuing) z rezervačních stanic do vykonávacích jednotek (execute). Po vykonání instrukce, tato vstupuje do dokončovacího buffru (completion/reorder buffer). Instrukce jsou v dokončovacím buffru dokončeny (opouštějí tento buffer) v programovém pořadí.



Literatura:

- Shen, J.P., Lipasti, M.H.: Modern Processor Design : Fundamentals of Superscalar Processors, First Edition, New York, McGraw-Hill Inc., 2005
- Brian Case: Intel Reveals Pentium Implementation Details, Vol. 7, No. 4, March 29, 1993