

ZÁKLADY SIGNÁLOVÝCH PROCESORŮ



Kurz: Signálové procesory

Autor: Petr Sysel

Lektor: Petr Sysel



Vytvoření této videopřednášky bylo podpořeno projektem č. CZ.1.07/2.2.00/28.0098
Evropského sociálního fondu a státním rozpočtem České republiky.

Obsah přednášky

Rozdíly mikrokontrolérů a signálových procesorů

Dělení do generací

Struktura Von Neumann

Harvardská architektura

- Vlastnosti architektury

- Aritmetická logická jednotka

- Instrukční soubor

Architektura LIW a VLIW

Superskalární architektura

Architektura paralelních systémů

Hlavní rozdíly mezi MCU a DSP?

	MCU	DSP
použití	řídící aplikace	zpracování signálů
architektura CPU	von Neumannova	Harvardská
základní formát	integer	fixed/fractional
saturace	pouze softwarově	automatická
přesnost	jednoduchá	rozšířený střadač
násobička	jednokvadrantová	plně čtyřkvadrantová
typická instrukce	read-modify-write	multiply and accumulate
adresovací režimy	bitové	bitově reverzní, modulo
fáze zřetězené zpracování	1-3	5-11

Dělení do generací

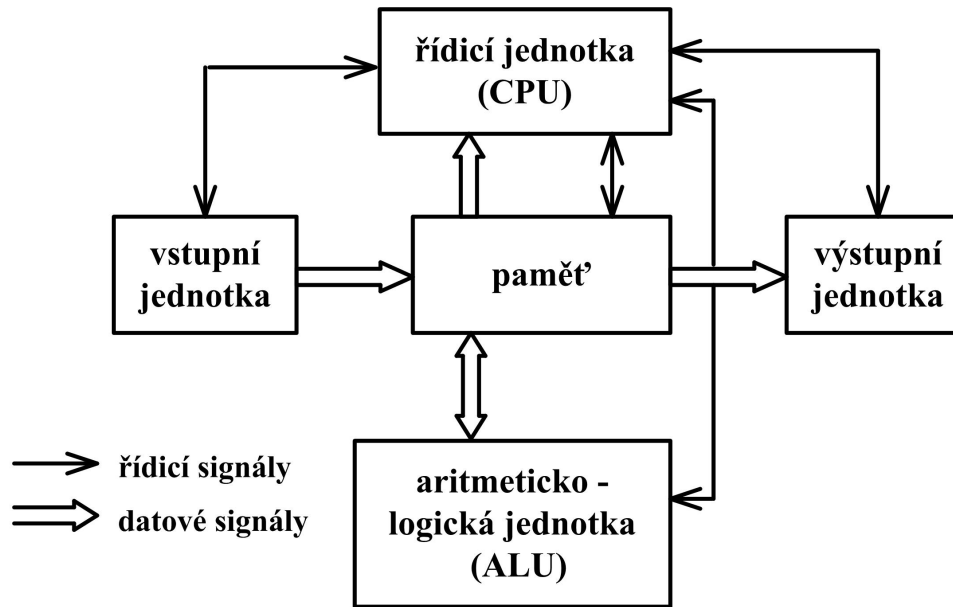
- 1. generace – 1980-1985
první signálové procesory, na čipu jsou již integrovány A/D a D/A převodníky, připojení vnějších periférií je však problematické stejně jako rozšiřování paměti, vzorkovací kmitočet je dán délkou vykonávání programu
i2920, NEC μ P7720, TMS320C10
- 2. generace – 1985-
procesory s pevnou řádovou čárkou, plně harvardská architektura, doplněny brány pro připojení periférií a vnější paměti, rozšíření o koprocesory
NEC μ P77220, TMS320C25, ADSP2100, DSP56000, DSP56300
- 3. generace –
procesory s pohyblivou řádovou čárkou, podporují standard IEEE754 pro vyjádření v pohyblivé řádové čárce
NEC μ P77230, TMS320C30, DSP96000, ADSP21000

Dělení do generací

- 4. generace – 2000-
procesory s pevnou nebo pohyblivou řádovou čárkou s paralelní architekturou LIW (Long Instruction Word) nebo VLIW (Very Long Instruction Word) nebo VLES (Variable Length Execution Set),
TMS320C6200, TMS320C6400, TMS320C6700, MSC8101,
- *future* – ?
vícejádrové signálové procesory – TMS320C6678 (8 jader), TMS320C6657,
...,
kombinované signálové procesory – kombinace jádra ARM a DSP –
TMS320DM8168, OMAP-L138, ...,
grafické procesory

Von Neumannova struktura

Navržena Johnem von Neumannem při vývoji počítače EDVAC - Electronic Discrete Variable Computer (1945).

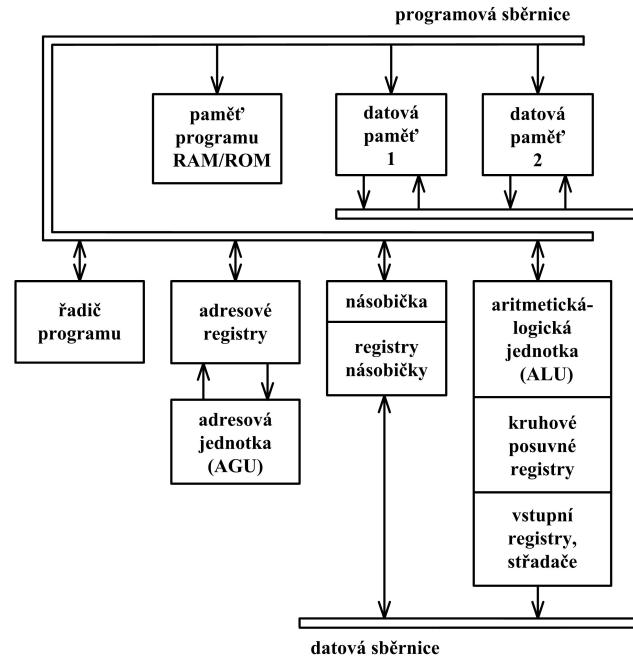


Von Neumannovy zásady

1. Funkční jednotky – paměť, řadič, aritmetická jednotka, vstupní a výstupní jednotka,
2. struktura je nezávislá na typu úlohy, programuje se obsahem paměti,
3. instrukce a operandy jsou v téže paměti,
4. paměť rozdělena do buněk stejné velikosti,
5. program tvoří instrukce, při změně dat se nemění,
6. změna pořadí instrukcí se vyvolá instrukcí skoku,
7. používají se dvojkové signály a dvojková číselná soustava.

Harvardská architektura

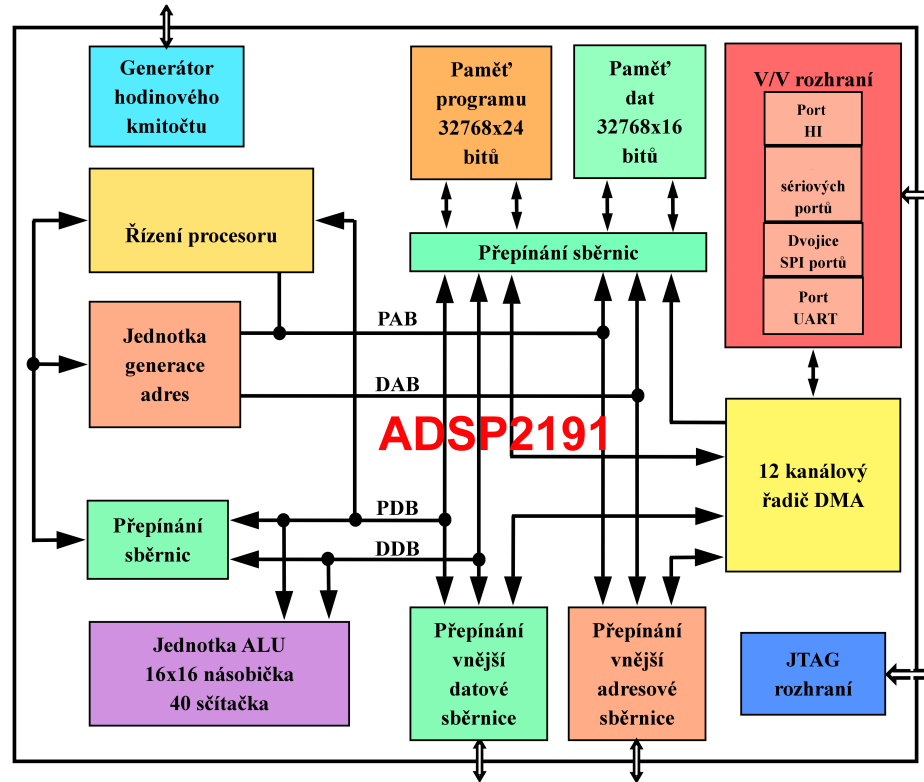
Navržena Howardem Aikenem na Harvardské univerzitě a použita pro elektronkový počítač ENIAC - Electronic Numerical Integrater and Calculator (1944).



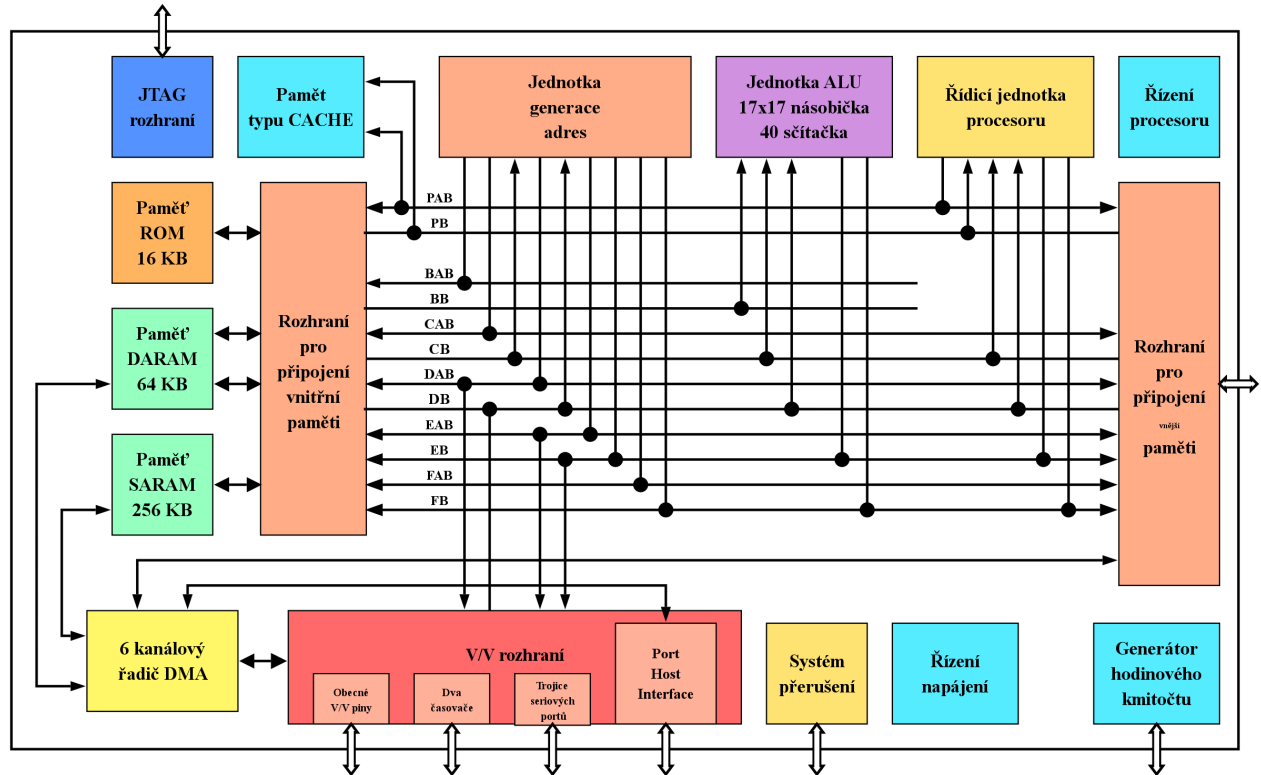
Harvardská architektura

1. Paměť je rozdělena na **paměť programu** a jednu nebo dvě **paměti dat** tak, aby současně dva operandy mohli vstupovat do ALU,
2. součin dvou operandů provádí v jednom cyklu **hardwarová násobička**,
3. pro zvýšení výpočetního výkonu se používá zřetězené zpracování instrukcí,
4. **pro současnou práci se dvěma operandy** je zvýšen počet adresových a datových sběrnic,
5. využívá se přímý přístup do paměti DMA,
6. řízení jádra je odděleno od řízení vstupních výstupních jednotek a periférií.

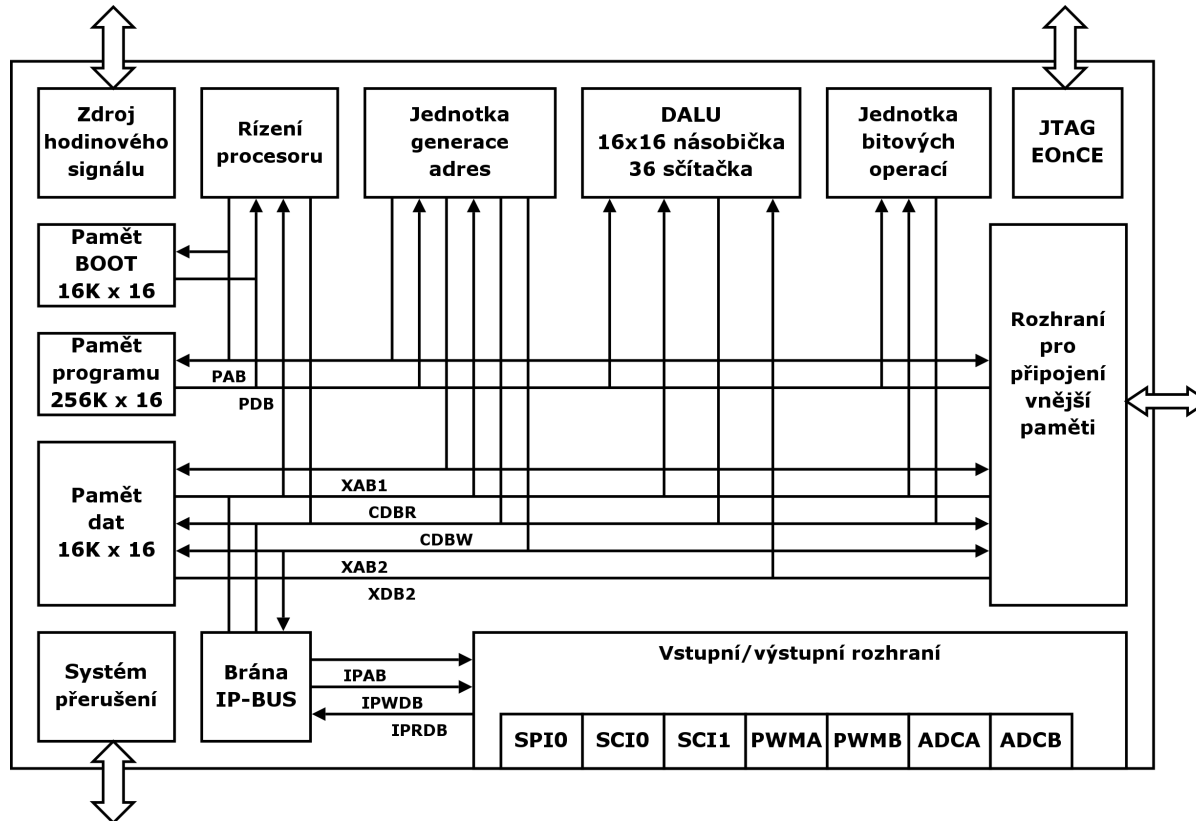
Analog Devices – ADSP2191



Texas Instruments – TMS320C5510



Freescape – 56F8367



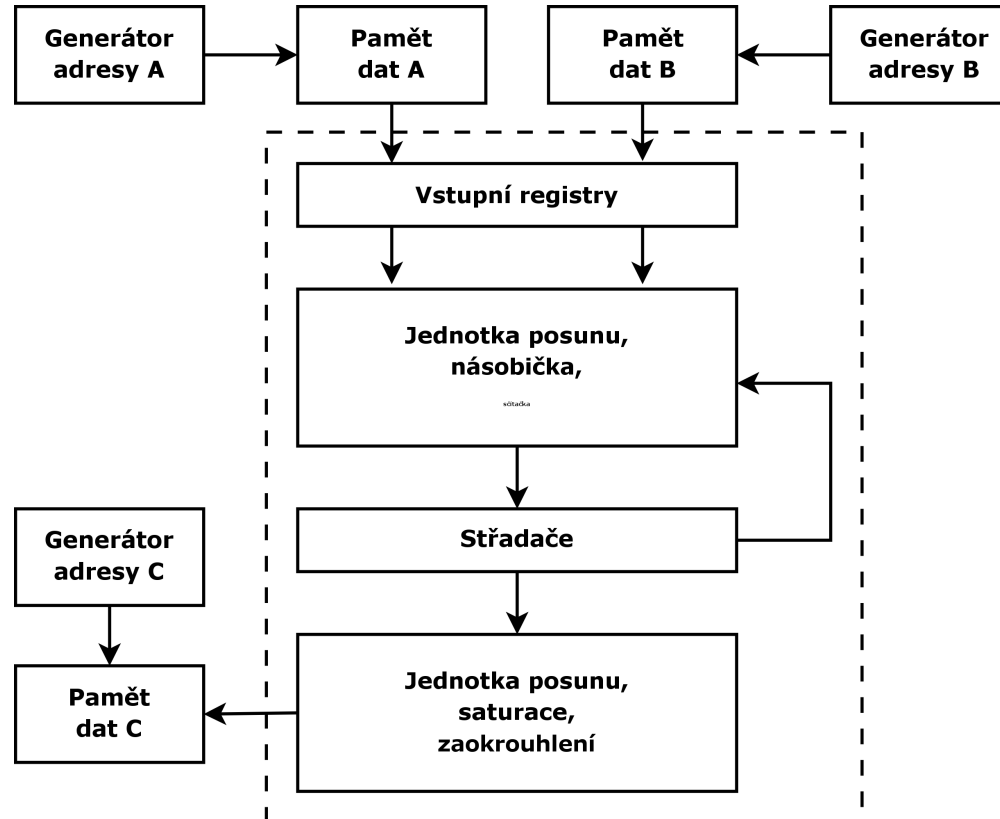
Zástupci signálových procesorů s harvardskou architekturou

	ADSP2191	TMS320C5510	56F8367
Max. kmitočet	160 MHz	200 MHz	60 MHz
Paměť programu	96 KB	256 KB SARAM	512 KB
Paměť dat	64 KB	64 KB DARAM	32 KB
BOOT ROM	3 KB	32 KB	32 KB
Délka slova	16 (program 24)	16 bitů	16 bitů
Rozhraní UART	1	3	0
Rozhraní SPI	2	lze použít UART	2
Rozhraní SCI	3	lze použít UART	2
Rozhraní HPI	1	1	0
Časovače	3	2	4
DMA kanálů	12	6	0

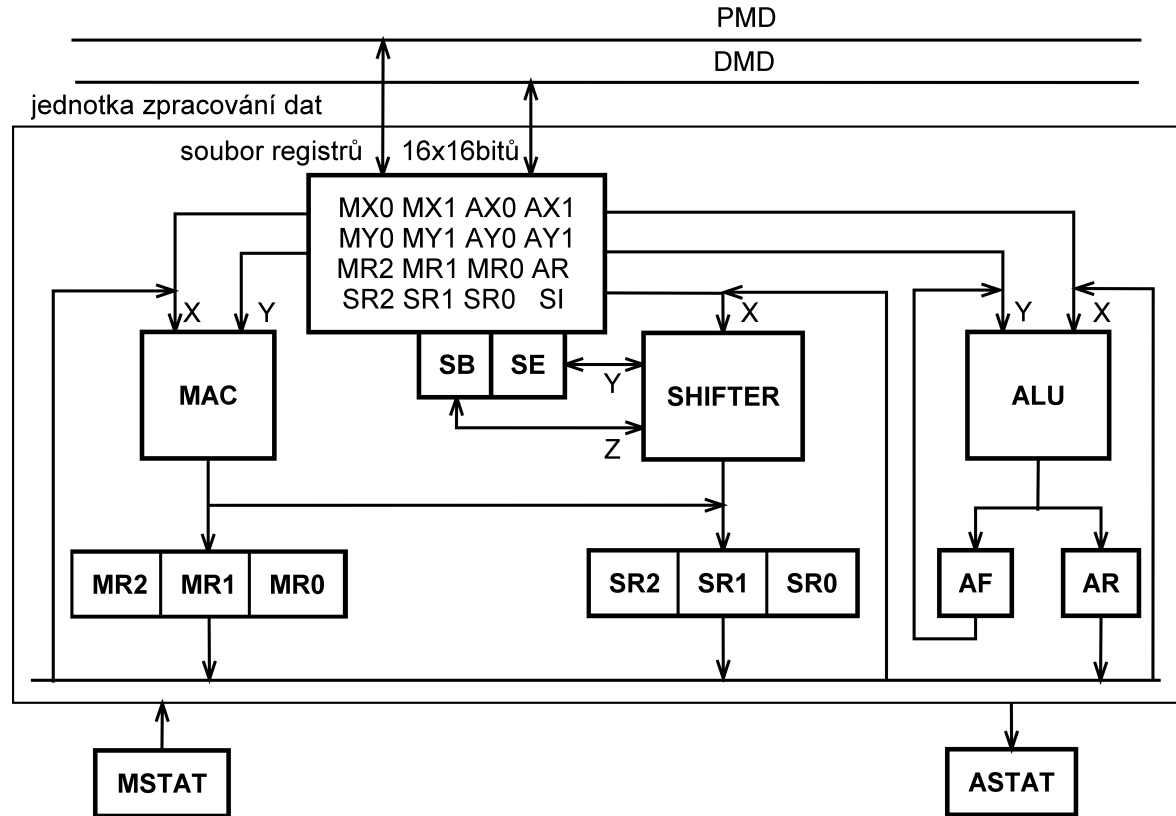
Pozn. Paměť u TMS320C5510 používá sjednocený paměťový prostor.

Aritmetická logická jednotka

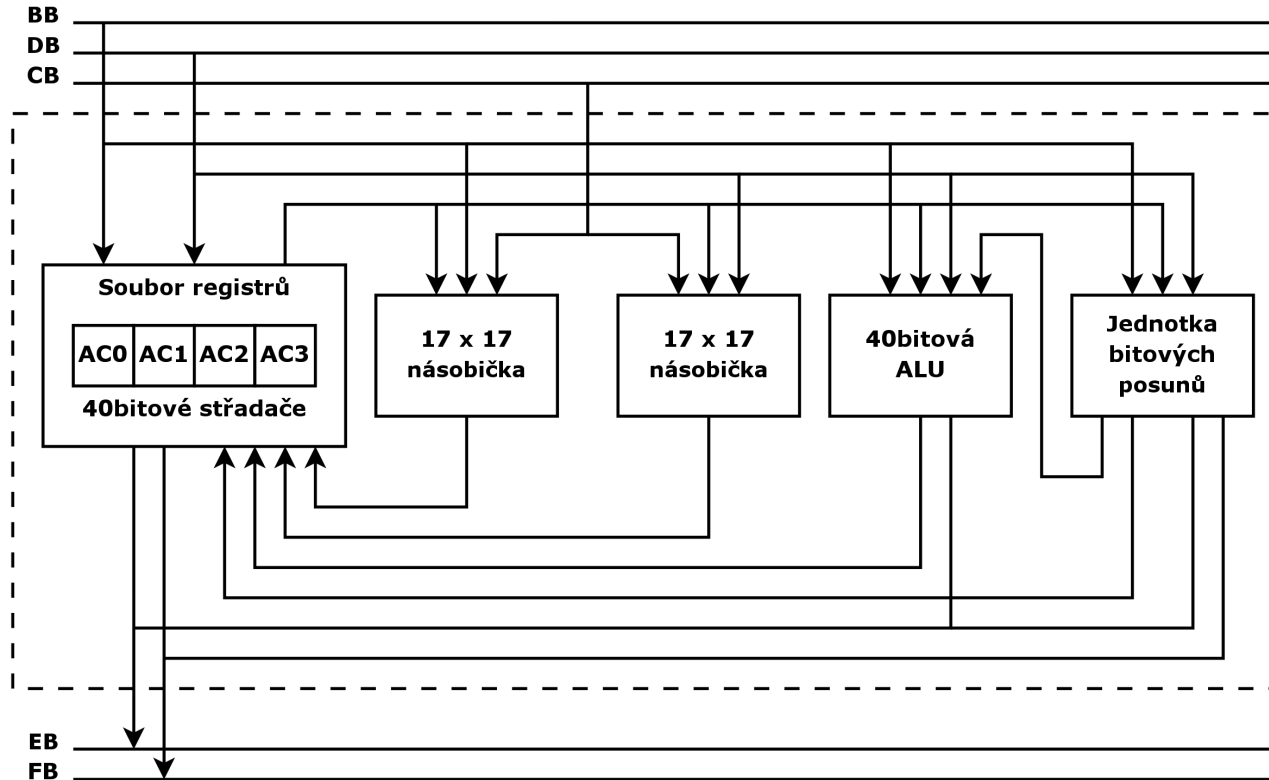
Jednotka je optimalizována pro operace násobení s akumulací.



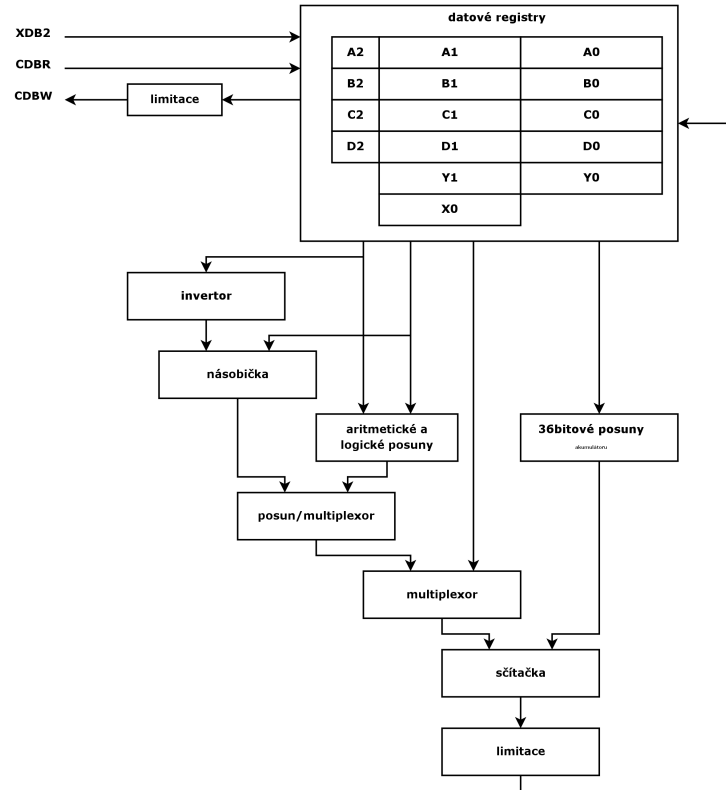
Aritmetická logická jednotka – ADSP2191



Aritmetická logická jednotka TMS320C5510



Aritmetická logická jednotka DSP56858



Typické vlastnosti instrukčního souboru

- Typická je instrukce násobení mac – multiply and accumulate, mac $x0, y0, a$, která provádí $a = a + x0 * y0$,
- typické jsou paralelní přesuny, např. u 56F8367:

návěští_instrukce_operandy_přesun1_přesun2_komentář
 mac $x0, y0, a$ $x: (r0)+, x0, x: (r3)+, y0$
 současně provádí operace:

- $a = a + x0 * y0$,
- přesun z paměti na adrese z registru $r0$ do registru $x0$,
- inkrementace adresy v registru $r0$,
- přesun z paměti na adrese z registru $r3$ do registru $y0$,
- inkrementace adresy v registru $r3$,
- podpora speciálních adresovacích režimu – modulo, bitově reverzní adresování,
- hardwarová podpora provádění cyklů.

Problémy instrukčního souboru

- Velký počet možných kombinací komplikuje kódování do strojového kódu,
- z důvodu zachování malých paměťových nároků (kompaktnosti) instrukčního souboru je používáno krátké instrukční slovo (16 bitů),
- krátké instrukční slovo zpětně komplikuje kódování typu instrukce a operandů,
- tatáž instrukce může být kódována různě v závislosti na operandech, přítomnosti paralelních přesunů, atd.,
- kvůli krátkému slovu nejsou povoleny všechny možné kombinace operandů a paralelních přesunů – instrukční soubor není **ortogonální**,
- to vše komplikuje zápis programu v assembleru, překlad i dekódovací jednotku řadiče programu.

Příklad kódování instrukce ABS

Procesor DSC 56F8367

Instruction Opcodes:

ABS F GGG,X:<ea_m>

15				12				11				8				7				4				3				0			
0	0	0	0	1	G	G	G	F	0	1	0	0	0	m	R	R															

ABS F X:<ea_m>,GGG

15				12		11	8			7	4			3	0	
0	0	1	0	1	G	G	G	F	0	1	0	0	m	R	R	

ABS FFF

15				12				11				8				7				4				3				0			
0	1	1	1	0	1	F	F	F	0	1	0	0	1	1	1																

Příklad kódování instrukce MAC

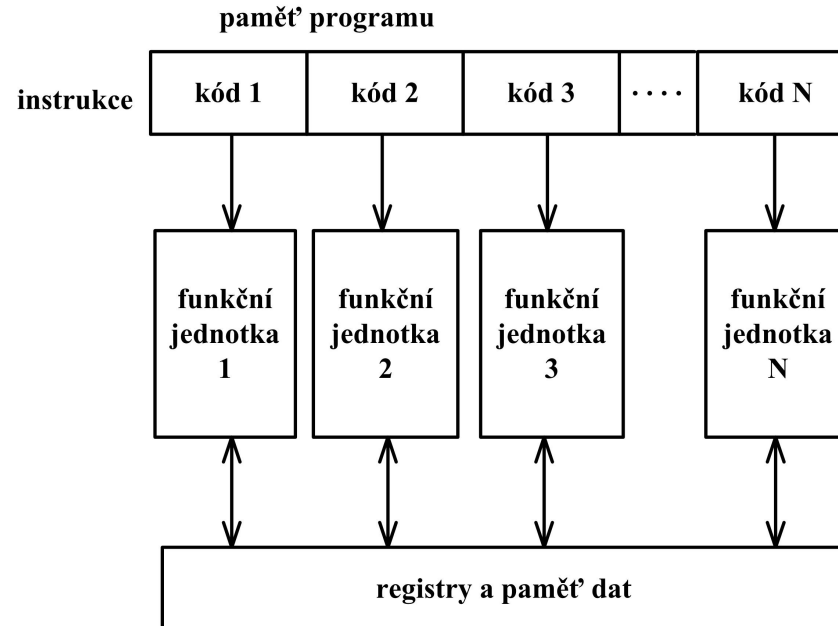
Procesor DSC 56F8367

Parallel Dual Reads:

Data ALU Operation ¹		First Memory Read		Second Memory Read	
Operation	Operands	Source 1	Destination 1	Source 2	Destination 2
MAC ²	Y1,X0,F Y1,Y0,F Y0,X0,F C1,Y0,F	X:(R0)+ X:(R0)+N X:(R1)+ X:(R1)+N	Y0 Y1	X:(R3)+ X:(R3)−	X0
		X:(R4)+ X:(R4)+N	Y0	X:(R3)+ X:(R3)+N3	X0
		X:(R0)+ X:(R0)+N X:(R4)+ X:(R4)+N	Y1	X:(R3)+ X:(R3)+N3	C

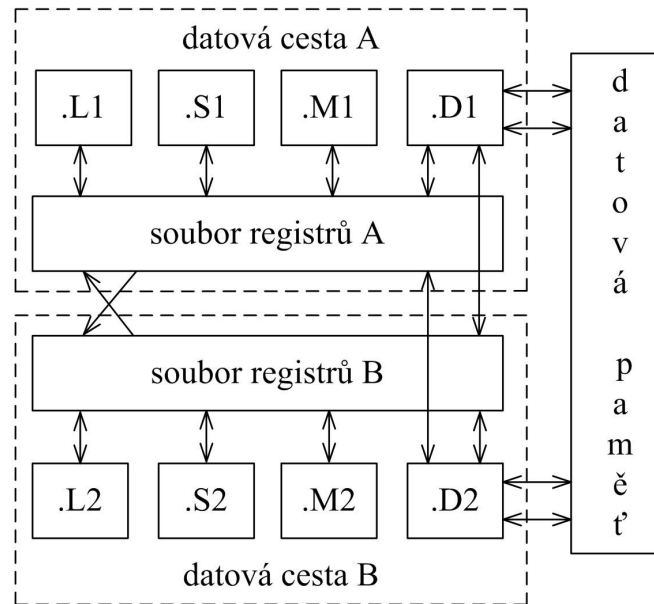
Architektura LIW a VLIW

Procesor obsahuje několik nezávislých jednotek, které jsou řízeny odděleně samostatnými instrukcemi sdruženými do jedné velké instrukce (**instrukčního paketu**).



Achitektura LIW a VLIW

Jádro signálového procesoru TMS320C6416 obsahuje 8 funkčních jednotek a může provádět až 8 instrukcí současně.



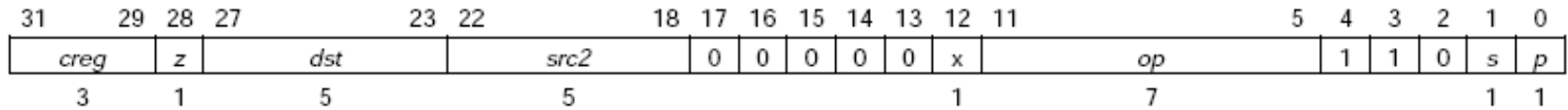
Vlastnosti instrukčního souboru

- Rozdělení instrukčního souboru na skupiny instrukcí podle funkčních jednotek,
- jednoduché kódování instrukcí, operandů i paralelního zpracování,
- téměř **ortogonální instrukční soubor** – instrukce mohou používat libovolný registr,
- proměnná délka instrukčního paketu umožňuje efektivní využití paměti.

Příklad kódování instrukce ABS

Procesor TMS320C6416

Opcode



Opcode map field used...	For operand type...	Unit	Optfield
<i>src2</i> <i>dst</i>	xsint sint	.L1, .L2	001 1010
<i>src2</i> <i>dst</i>	slong slong	.L1, L2	011 1000

Příklad kódování instrukce SMPY

Procesor TMS320C6416

Opcode

31	29	28	27	23	22	18	17	13	12	11	10	9	8	7	6	5	4	3	2	1	0
<i>creg</i>		<i>z</i>		<i>dst</i>		<i>src2</i>		<i>src1</i>		<i>x</i>	1	1	0	1	0	0	0	0	0	<i>s</i>	<i>p</i>
3		1		5		5		5		1										1	1

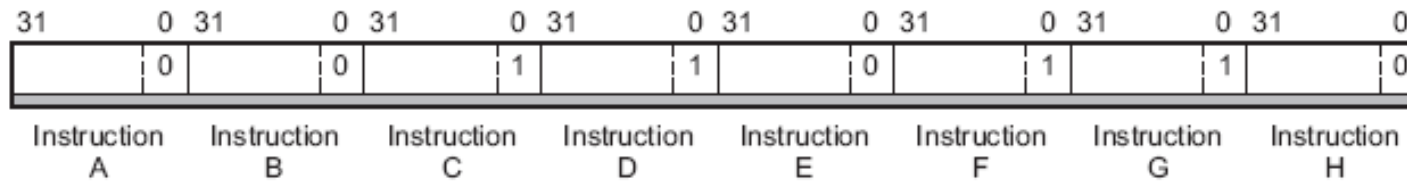
Opcode map field used...	For operand type...	Unit
<i>src1</i>	slsb16	.M1, .M2
<i>src2</i>	xslsb16	
<i>dst</i>	sint	

Příklad kódování instrukčních paketů

Procesor TMS320C6416

Example 3-3. Partially Serial p-Bit Pattern in a Fetch Packet

This p -bit pattern:



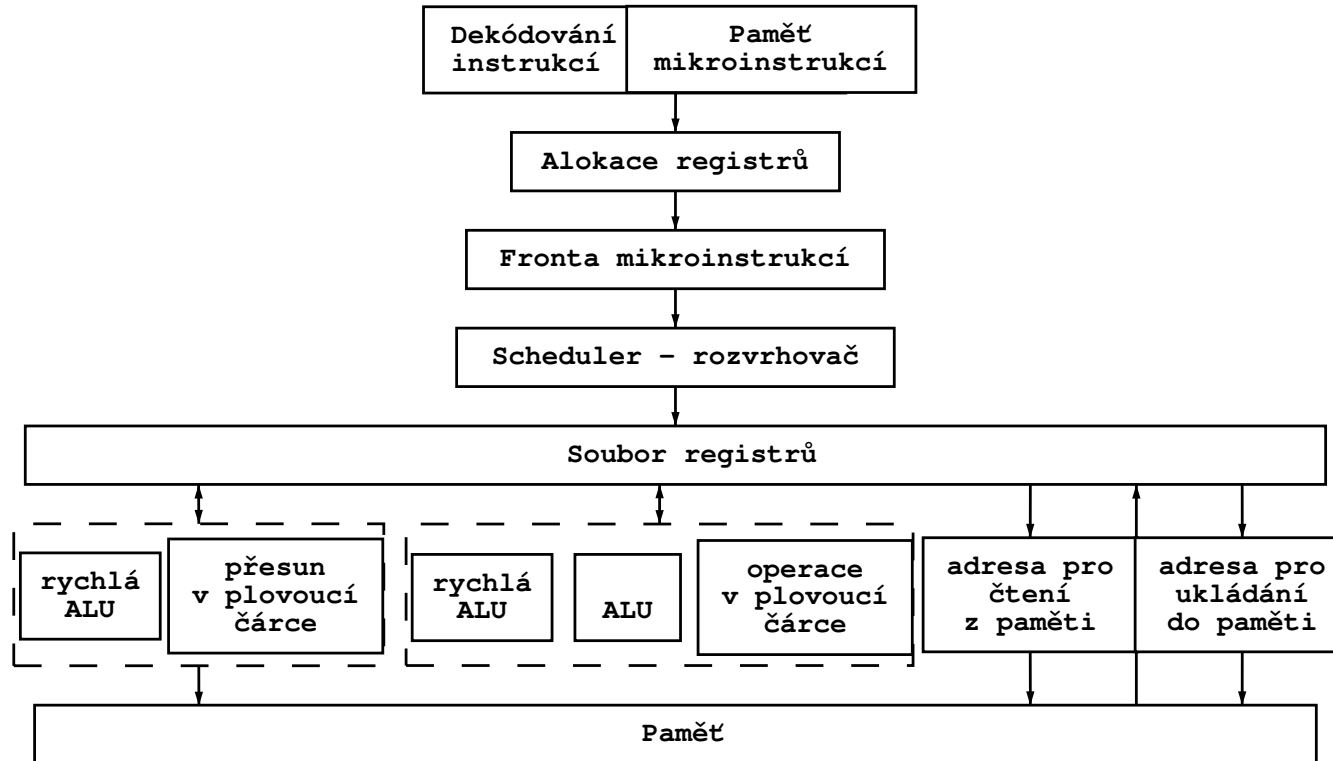
results in this execution sequence:

Cycle/Execute Packet	Instructions		
1	A		
2	B		
3	C	D	E
4	F	G	H

Superskalární architektura

- Podobně jako architektura VLIW využívá paralelně pracující funkční jednotky.
- Rozdělování instrukcí mezi funkční jednotky řídí speciální hardwarová jednotka (scheduler) za běhu programu.
 - binární kompatibilita,
 - složitost rozvrhovače,
 - obtížná optimalizace,
- Nejznámějším představitelem je procesor INTEL PENTIUM.

Superskalární architektura



Architektura paralelních systémů

Podle současně zpracovávaných instrukcí rozlišujeme systémy:

- SI – Single Instruction – jeden zpracovávaný proud instrukcí,
- MI – Multiple Instructions – více zpracovávaných proudů instrukcí.

Podle současně zpracovávaných datových proudů rozlišujeme systémy:

- SD – Single Data – jeden zpracovávaný proud dat,
- MD – Multiple Data – více zpracovávaných proudů dat.

Vzájemnou kombinací vzniknou čtyři možnosti:

- SISD – klasický von Neumannův počítač,
- SIMD – jeden proud instrukcí zpracovává více datových proudů (např. MMX u Pentia, LIW, VLIW, VLES architektury),
- MISD – hypotetická kombinace, kdy několik proudů instrukcí zpracovává jeden datový proud,
- MIMD – obecný typ paralelního systému, kdy několik proudů instrukcí zpracovává několik proudů dat.

Paralelní zpracování dat SIMD

Využívá např. rozšíření instrukční sady MMX procesoru Pentium nebo signálové procesory řady TMS320C6400.

