

Kódování instrukcí

Úvod

Instrukce musí obsahovat:

- typ operace
- adresu operandu (operandů)
- typ operandů
- modifikátory adresy
- modifikátory operace

Pevná a proměnná délka instrukce (1)

Pevná délka instrukce

- + snadné a rychlé čtení a dekódování
- nízká efektivita kódování

Příklad: Alpha

31	26 25	21 20	16 15	5 4	0	
Opcode	Number					PALcode Format
Opcode	RA	Disp				Branch Format
Opcode	RA	RB	Disp			Memory Format
Opcode	RA	RB	Function	RC		Operate Format

Instrukční soubor procesoru Alpha

Aritmetické a logické operace:

- Jen mezi registry.
- Třídresové instrukce (**Operate Format**).
 - např. **ADDL R1,R2,R3** ($R3 \leftarrow R1 + R2$).

Adresování paměti:

- Vždy nepřímo pomocí některého registru.
- Adresa ve tvaru **@(RB + Displacement)** (**Memory format**).
 - např. **LDL R1,0x0100(R2)** ($R1 \leftarrow @(R2 + 0x0100)$)

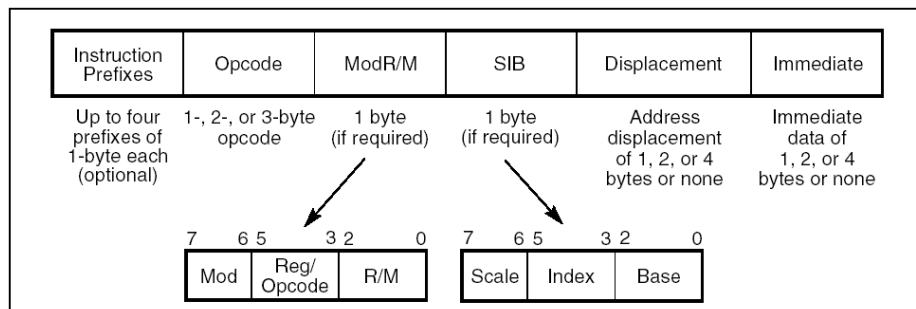
31	26 25	21 20	16 15	5 4	0	
Opcode	Number					PALcode Format
Opcode	RA	Disp				Branch Format
Opcode	RA	RB	Disp			Memory Format
Opcode	RA	RB	Function	RC		Operate Format

Pevná a proměnná délka instrukce (2)

Proměnná délka instrukce

- obtížné čtení a dekodování
- + efektivnější kódování složitých instrukcí

Příklad: IA-32



IA-32 Instruction Format

Instrukční soubor IA-32

Aritmetické a logické operace

Operandy mohou být v registrech nebo v paměti.

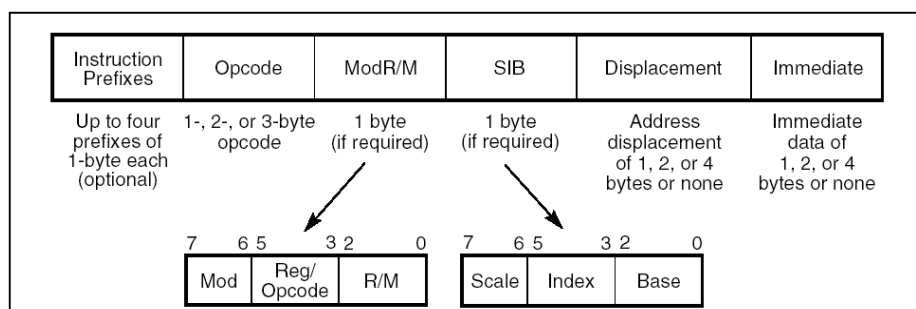
Dvouadresové instrukce.

např. `ADD EAX, 0x0100` ($EAX \leftarrow EAX + @0x0100$).

Adresování paměti

Adresa může být např.:

`MOV EAX, EBX+[ECX*4]+0x0100` ($EAX \leftarrow @(EBX + ECX*4 + 0x0100)$).

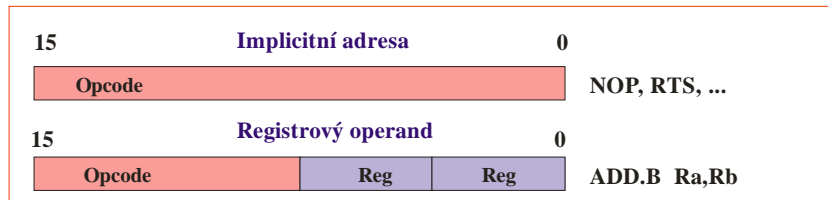


IA-32 Instruction Format

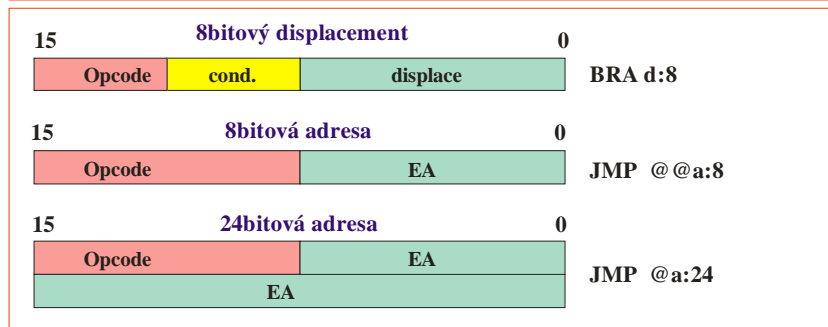
Kódování instrukcí procesoru H8S (1)

- Délka instrukcí je 2, 4, 6 nebo 8 bytů.
- Při adresování instrukcí (skoky atd.) je délka adresy 8 nebo 24 bitů.

Implicitní a
registrový operand



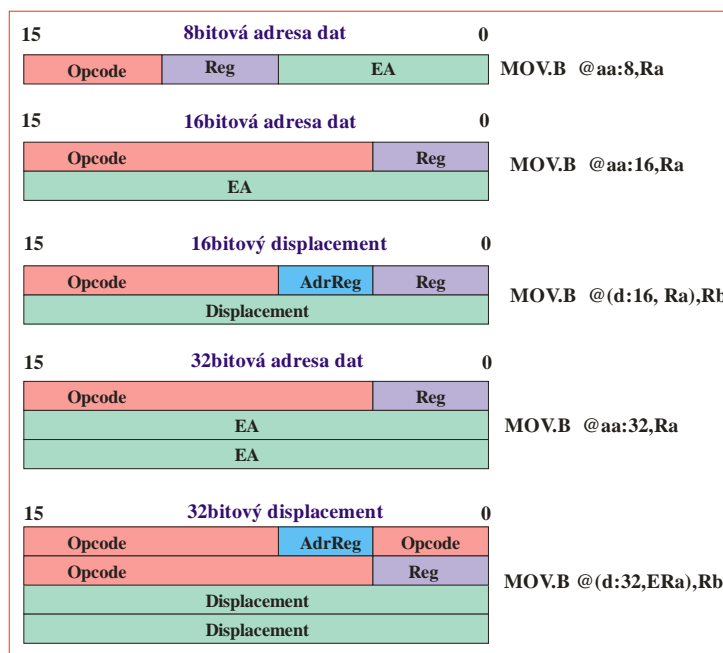
Adresování
instrukcí



Kódování instrukcí procesoru H8S (2)

- Délka instrukcí je 2, 4, 6 nebo 8 bytů.
- Při adresování dat je délka adresy je 8, 16 nebo 32 bitů.

Adresování dat



Příklad: Instrukce MOV.W

MOV.W <EA>, Rd

Operand Format and Number of States Required for Execution

Addressing Mode	Mnemonic	Operands	Instruction Format								No. of States		
			1st byte	2nd byte	3rd byte	4th byte	5th byte	6th byte	7th byte	8th byte			
Immediate	MOV.W	#xx:16, Rd	7	9	0	rd	IMM						2
Register indirect	MOV.W	@ERs, Rd	6	9	0	ers	rd						2
Register indirect with displacement	MOV.W	@(d:16, ERs), Rd	6	F	0	ers	rd	disp					3
	MOV.W	@(d:32, ERs), Rd	7	8	0	ers	0	6	B	2	rd	disp	
Register indirect with post-increment	MOV.W	@ERs+, Rd	6	D	0	ers	rd						3
Absolute address	MOV.W	@aa:16, Rd	6	B	0	rd	abs						3
	MOV.W	@aa:32, Rd	6	B	2	rd	abs						4

Logický a fyzický adresní prostor

Logický adresní prostor

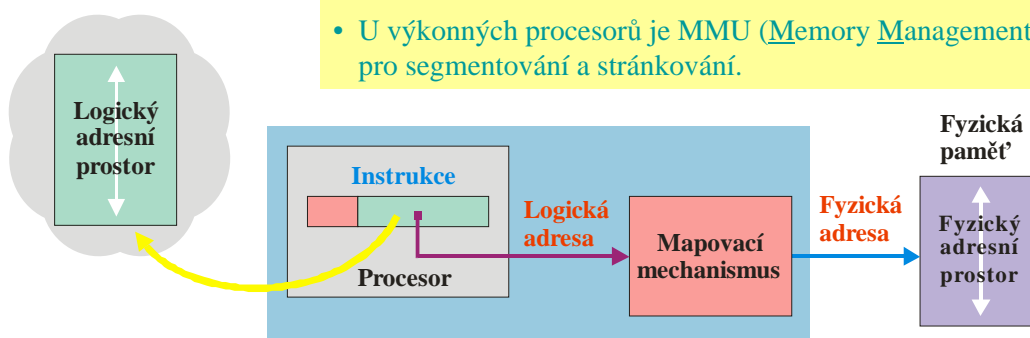
- Je určen rozsahem adres, které může procesor zpracovat (délka adres v instrukci + u některých procesorů modifikace adres segmentováním).

Fyzický adresní prostor

- Je určen velikostí fyzické paměti, se kterou může procesor pracovat (počet adresních vodičů).

Mapovací mechanismus:

- U jednoduchých procesorů není (logický a.p. = fyzický a.p.).
- U výkonných procesorů je MMU (Memory Management Unity) pro segmentování a stránkování.



Formát adresy instrukcí procesoru H8S

Logické adresování instrukcí používá 24bitovou adresu.

- V instrukci lze použít 8 nebo 24bitovou adresu.
- Vyšší adresní bity doplní procesor.



Převod logické adresy na fyzickou

- Logická adresa je 32bitová (data) nebo 24bitová (instrukce).
- Fyzická adresa je 24bitová.
- 8 nejvyšších bitů adresy dat se ignoruje.

