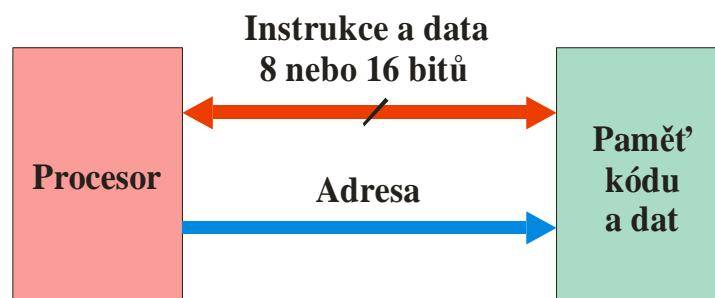


Provádění instrukcí procesorem

Základní model

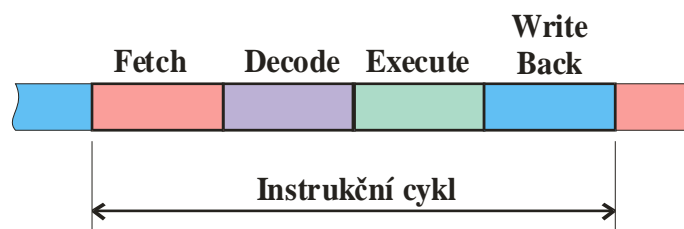
- Kód programu (instrukce) a data jsou uloženy ve vnější paměti.
- Procesor musí nejprve z paměti přečíst instrukci.
- Při provedení instrukce podle potřeby čte nebo zapisuje data z/do paměti.

– H8S: Každé čtení nebo zápis trvá 3 takty hodin procesoru.



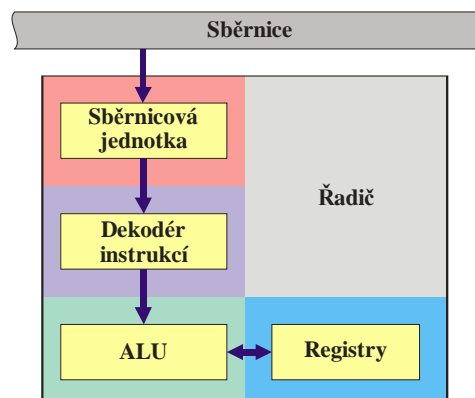
Fáze instrukčního cyklu (1)

- Instrukční cykl = doba zpracování jedné instrukce.
- 4 základní fáze:
 - **Fetch** – čtení kódu instrukce z paměti,
 - **Decode** – dekódování instrukce,
 - **Execute** – provedení instrukce,
 - **Write Back** – zpětný zápis výsledku do registrů procesoru.
- Podle typu instrukce mohou být jednotlivé fáze různě dlouhé – vyžadují různý počet **strojových cyklů**.



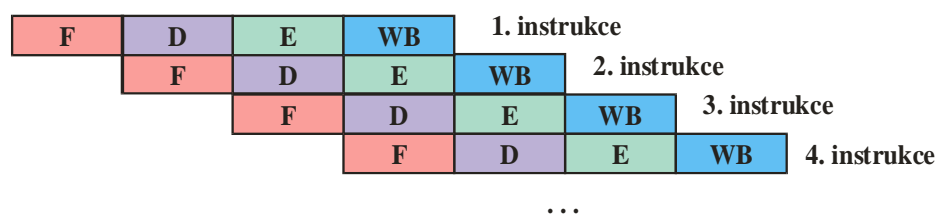
Fáze instrukčního cyklu (2)

- 4 základní fáze:
 - **Fetch** – čtení kódu instrukce z paměti,
 - **Decode** – dekódování instrukce,
 - **Execute** – provedení instrukce,
 - **Write Back** – zpětný zápis výsledku do registrů procesoru.
- Každou fází instrukce vykonává jiná část CPU.



Proudové zpracování instrukcí

- Při sekvenčním zpracování je využita vždy jen část CPU.
- Proudové zpracování (pipeline) : v CPU se zpracovává několik instrukcí, každá v jiné fázi.
- Požadavky:
 - stejná délka zakódovaných instrukcí,
 - stejná délka provádění instrukcí.
- **RISC** – vyhovuje ??
- **CISC** – nevyhovuje ??



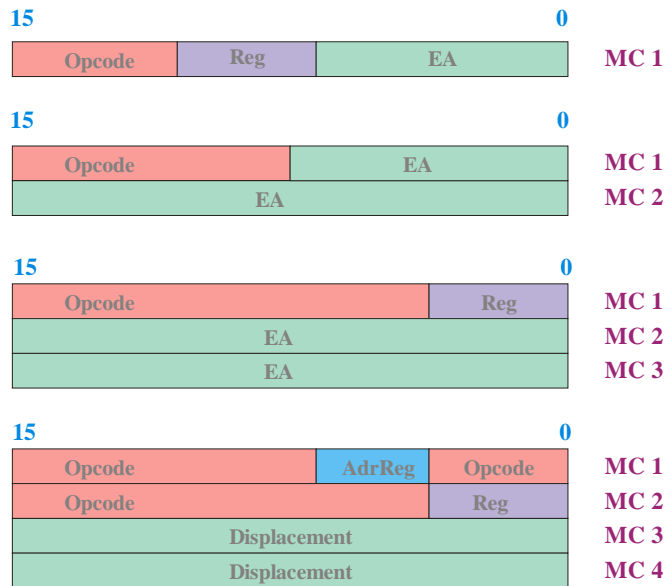
CISC vs. RISC

- **CISC:**
 - Složitě instrukce \Rightarrow rozsáhlý instrukční soubor.
 - Různá délka instrukcí.
 - Proměnná délka instrukčního cyklu.
 - + Účinné instrukce \Rightarrow program obsahuje malé množství instrukcí.
 - Obtížné proudové zpracování \Rightarrow instrukce se provádí pomalu.
- **RISC:**
 - Jednoduché instrukce \Rightarrow omezený instrukční soubor.
 - Instrukce jsou stejné délky.
 - Konstantní délka instrukčního cyklu.
 - Málo účinné instrukce \Rightarrow program obsahuje velké množství instrukcí.
 - + Snadné proudové zpracování \Rightarrow instrukce se provádí rychle.

Zpracování instrukcí v procesoru H8S (1)

Čtení kódu instrukce:

- provádí se postupně po 16 bitech ⇒ každé čtení = 1 MC (strojový cykl),
- doba 1 MC závisí i na šířce datové sběrnice (3 nebo 6 taktů hodin).



Zpracování instrukcí v procesoru H8S (2)

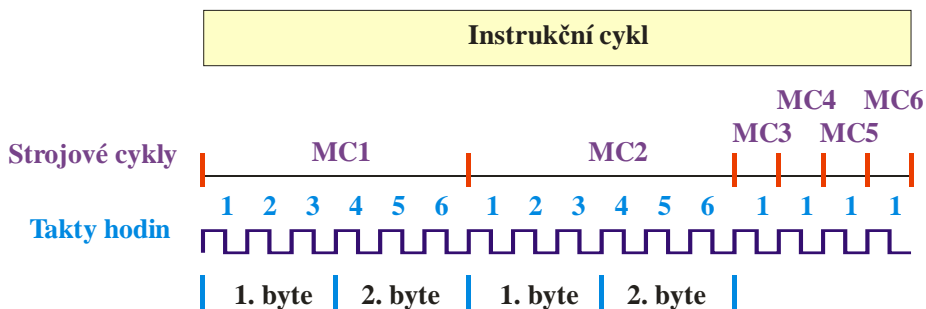
Čtení kódu instrukce:

- provádí se postupně po 16 bitech ⇒ každé čtení = 1 MC (strojový cykl),
- doba 1 MC závisí i na šířce datové sběrnice (3 nebo 6 taktů hodin).

Dekódování, provedení a zpětný zápis:

- vyžaduje různý počet (1 – 19) MC,
- každý MC trvá 1 takt hodin nebo 3/6 taktů při práci s pamětí,
- současně se provádí čtení další instrukce ⇒ MC potom trvá více taktů hodin.

Příklad:



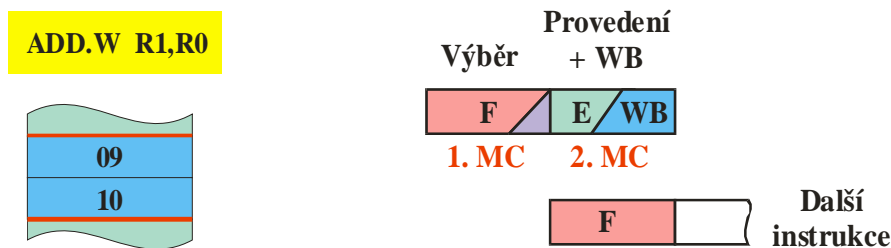
Zpracování instrukcí v procesoru H8S (3)

Čtení kódu instrukce:

- provádí se postupně po 16 bitech ⇒ každé čtení = 1 MC (strojový cykl),
- doba 1 MC závisí i na šířce datové sběrnice (3 nebo 6 taktů hodin).

Dekódování, provedení a zpětný zápis:

- vyžaduje různý počet (1 – 19) MC,
- každý MC trvá 1 takt hodin nebo 3/6 taktů při práci s pamětí,
- současně se provádí čtení další instrukce ⇒ MC potom trvá více taktů hodin.



Instrukce ADD.W

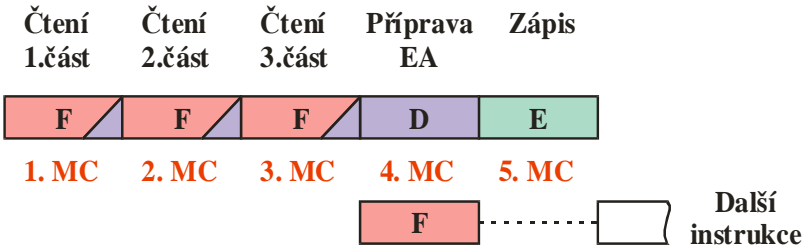
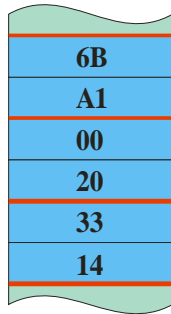
Operand Format and Number of States Required for Execution

Addressing Mode	Mnemonic	Operands	Instruction Format				No. of States	
			1st byte	2nd byte	3rd byte	4th byte		
Immediate	ADD.W	#xx:16, Rd	7	9	1	rd	IMM	2
Register direct	ADD.W	Rs, Rd	0	9	rs	rd		1

Instruction	1	2	
ADD.B #xx:8,Rd	R:W NEXT		
ADD.B Rs,Rd	R:W NEXT		
ADD.W #xx:16,Rd	R:W 2nd	R:W NEXT	
ADD.W Rs,Rd	R:W NEXT		
ADD.L #xx:32,ERd	R:W 2nd	R:W 3rd	R:W NEXT
ADD.L ERs,ERd	R:W NEXT		
ADDS #1/2/4,ERd	R:W NEXT		
ADDX #xx:8,Rd	R:W NEXT		

Zpracování instrukcí v procesoru H8S (4)

MOV.W R1,@00203314



Instrukce MOV.W

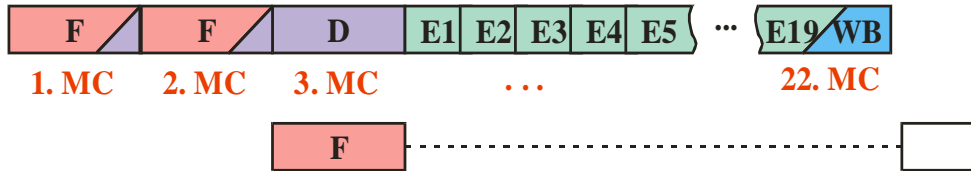
Operand Format and Number of States Required for Execution

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			
Register indirect	MOV.W	Rs, @ERd	6	9	1:erd: rs								2
Register indirect with displacement	MOV.W	Rs, @(d:16, ERd)	6	F	1:erd: rs	disp							3
	MOV.W	Rs, @(d:32, ERd)	7	8	0:erd: 0	6	B	A	rs	disp			5
Register indirect with pre-decrement	MOV.W	Rs, @-ERd	6	D	1:erd: rs								3
Absolute address	MOV.W	Rs, @aa:16	6	B	8	rs	abs						3
	MOV.W	Rs, @aa:32	6	B	A	rs	abs					4	

Instruction	1	2	3	4	5
MOV.W Rs,@(d:16,ERd)	R:W 2nd	R:W NEXT	W:W EA		
MOV.W Rs,@(d:32,ERd)	R:W 2nd	R:W 3rd	R:E 4th	R:W NEXT	W:W EA
MOV.W Rs,@aa:16	R:W 2nd	R:W NEXT	W:W EA		
MOV.W Rs,@aa:32	R:W 2nd	R:W 3rd	R:W NEXT	W:W EA	
MOV.W Rs,@-ERd	R:W NEXT	Internal operation, 1 state		W:W EA	
MOVL #xx:32,ERd	R:W 2nd	R:W 3rd	R:W NEXT		
MOVL ERs,ERd	R:W NEXT				
MOVL @ERs,ERd	R:W 2nd	R:W:M NEXT	R:W:M EA	R:W EA+2	
MOVL @(d:16,ERs),ERd	R:W 2nd	R:W:M 3rd	R:W NEXT	R:W:M EA	R:W EA+2
MOVL @(d:32,ERs),ERd	R:W 2nd	R:W:M 3rd	R:W:M 4th	R:W 5th	R:W NEXT
MOVL @ERs+,ERd	R:W 2nd	R:W:M NEXT	Internal operation	R:W:M EA	R:W EA+2

Zpracování instrukcí v procesoru H8S (5)

DIVXS.W R1,ER0



Instrukce DIVXS.W

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			
Register direct	DIVXS.W	Rs, ERd	0	1	D	0	5	3	rs	0	erd	21

DAS Rd	R:W NEXT										
DEC.B Rd	R:W NEXT										
DEC.W #1/2,Rd	R:W NEXT										
DEC.L #1/2,ERd	R:W NEXT										
DIVXS.B Rs,Rd	R:W 2nd	R:W NEXT	Internal operation, 11 states								
DIVXS.W Rs,ERd	R:W 2nd	R:W NEXT	Internal operation, 19 states								
DIVXU.B Rs,Rd	R:W NEXT	Internal operation, 11 states									
DIVXU.W Rs,ERd	R:W NEXT	Internal operation, 19 states									
EEPMOV.B	R:W 2nd	R:B EAs *1	R:B EAd *1	R:B EAs *2	W:B EAd *2	R:W NEXT					
EEPMOV.W	R:W 2nd	R:B EAs *1	R:B EAd *1	R:B EAs *2	W:B EAd *2	R:W NEXT					
EXTS.W Rd	R:W NEXT			← Repeated n times*3 →							
EXTS.L ERd	R:W NEXT										
EXTU.W Rd	R:W NEXT										