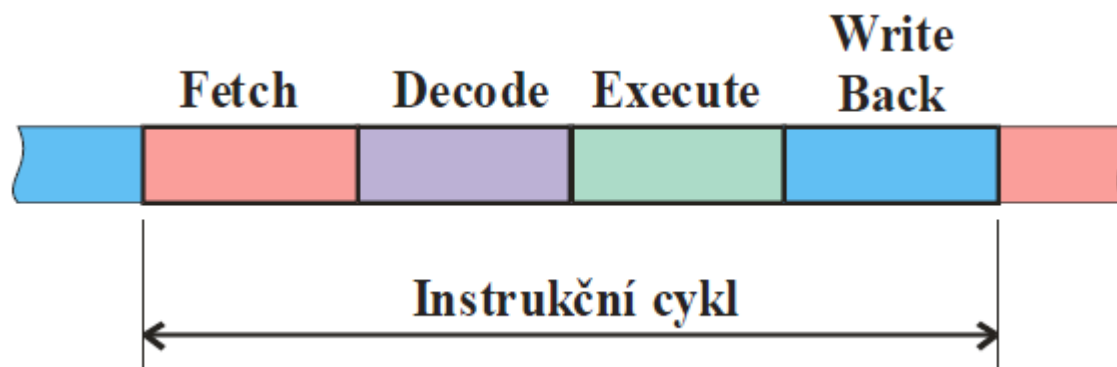


strojový cyklus

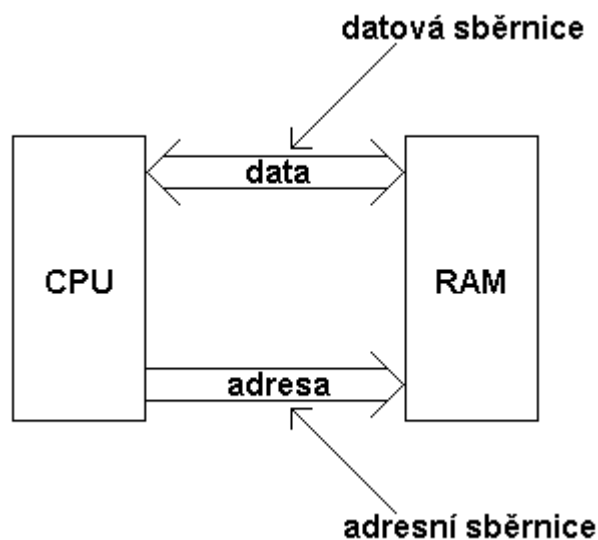
Při provádění instrukce musí procesor nejdříve načíst příkaz z paměti, dekodovat ho, provést daný příkaz a zapsat výsledek. Ne každá instrukce musí mít všechny části a ne každá instrukce musí potřebovat vždy stejný počet taktů, na provedení instrukce.



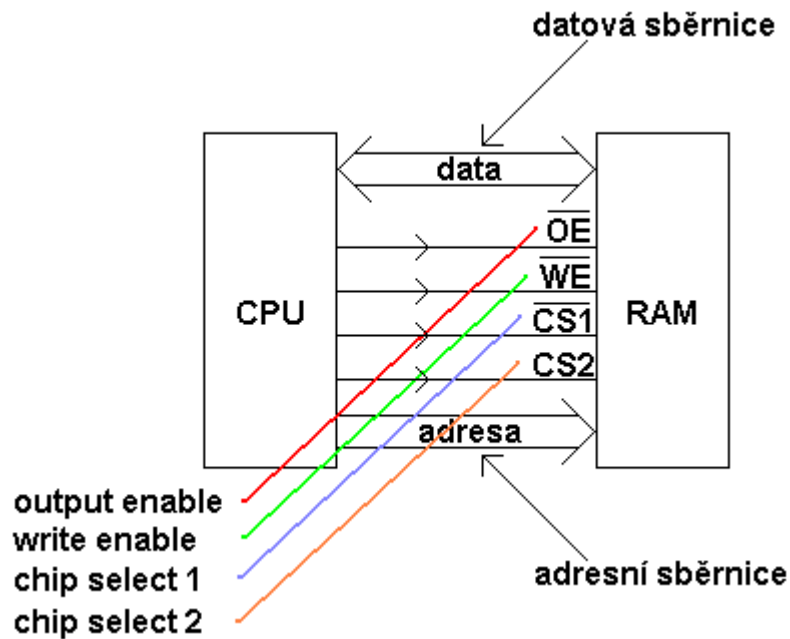
Rychlost přečtení instrukce závisí na tom, jestli se čte z interní, nebo externí paměti a také na tom, jestli se čte po 8, 16 či 32 bitech. H8S potřebuje tři takty hodin procesoru k tomu, aby přečetla, či uložila data z/do paměti.

Sběrnice

Program i data, která procesor potřebuje ke své činnosti, jsou uložena v paměti (buď externí, nebo interní). Pro načtení dat potřebuje procesor paměti sdělit adresu, kde se požadovaná informace nachází, aby mu paměť mohla data „sdělit“, viz obrázek.



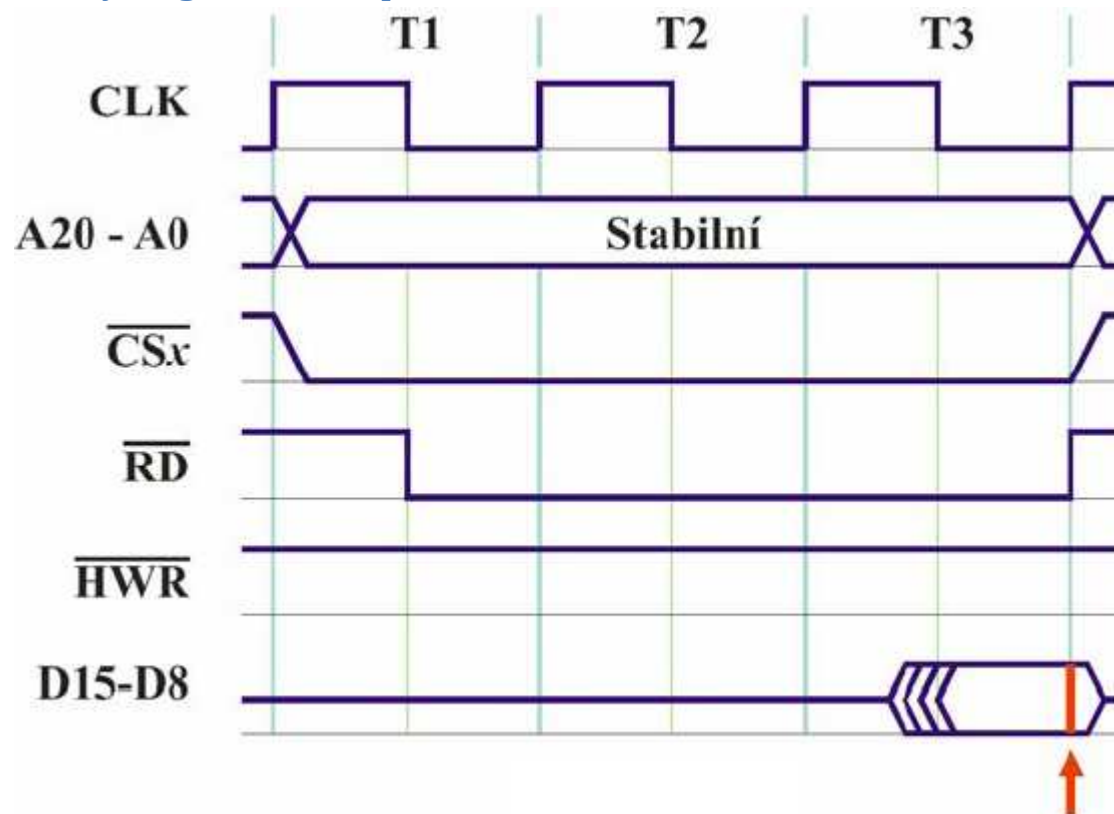
K přenosu dat z/do paměti potřebuje počítač datovou a adresní sběrnici. Aby ovšem paměť věděla, že počítač s ní komunikuje a jestli má data přečíst, nebo zapsat, potřebuje ještě další pomocné vodiče.



- **!OE** (output enable) povel, aby paměť „okopírovala“ na datovou sběrnici obsah paměti z adresy nastavené adresní sběrnici.
- **!WE** (write enable) povel, aby paměť zapsala data z datové sběrnice do své paměti na adresu danou adresní sběrnici
- **!CS1** (Chip select 1) pokud je v logické jedničce, tak uvede paměť do neaktivního stavu a tak můžou na sběrnici pracovat jiná zařízení, např. jiná paměť.
- **CS2** (Chip select 2) pokud je v logické nule, tak uvede paměť do neaktivního stavu a tak můžou na sběrnici pracovat jiná zařízení, např. jiná paměť.

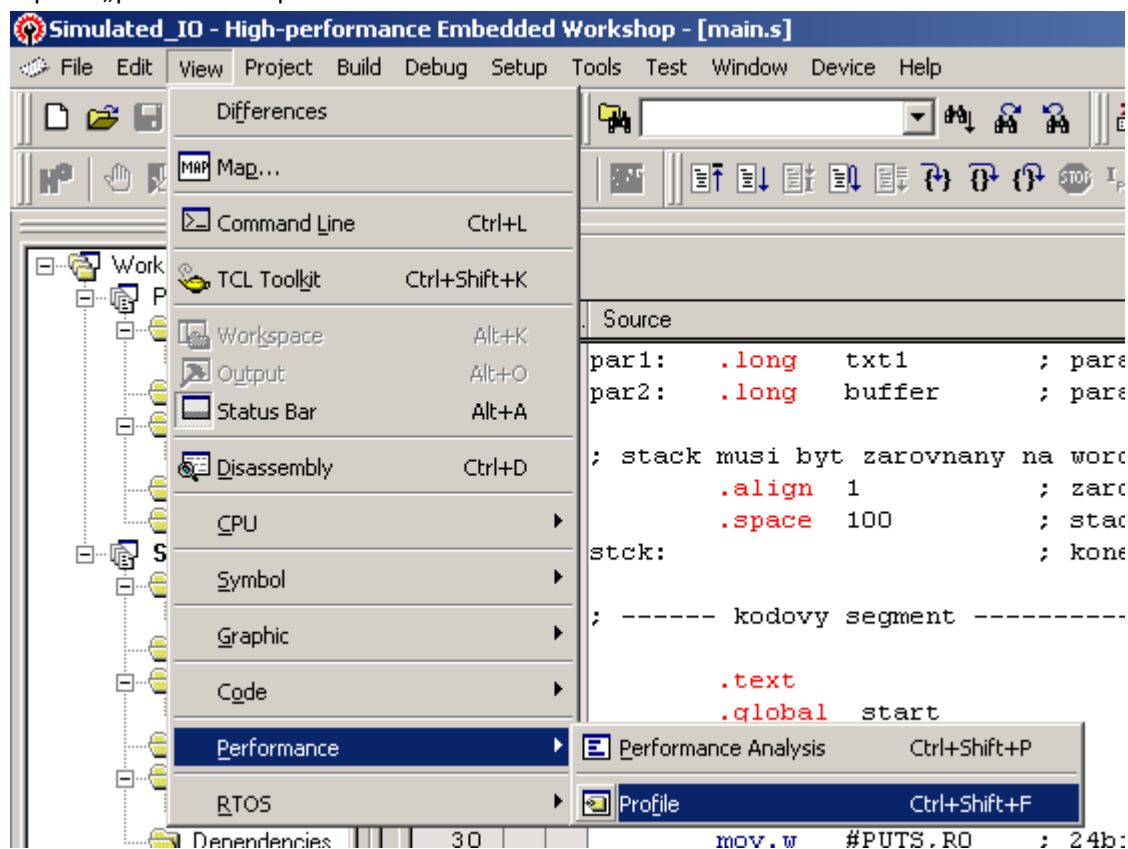
Pozn. Označení OE, WE... se může lišit podle výrobce paměti, jak si tyto vývody pojmenuje (např. RE(read enable), HWR apod.)

Časový diagram čtení z paměti



Vývojové prostředí HEW umožňuje analyzovat program z hlediska počtu přístupů do paměti, pokud budeme v nastavení měnit šířku sběrnice a počet taktů pro přístup do paměti, můžeme pak sledovat změny ve výkonu (potřebného počtu taktů).

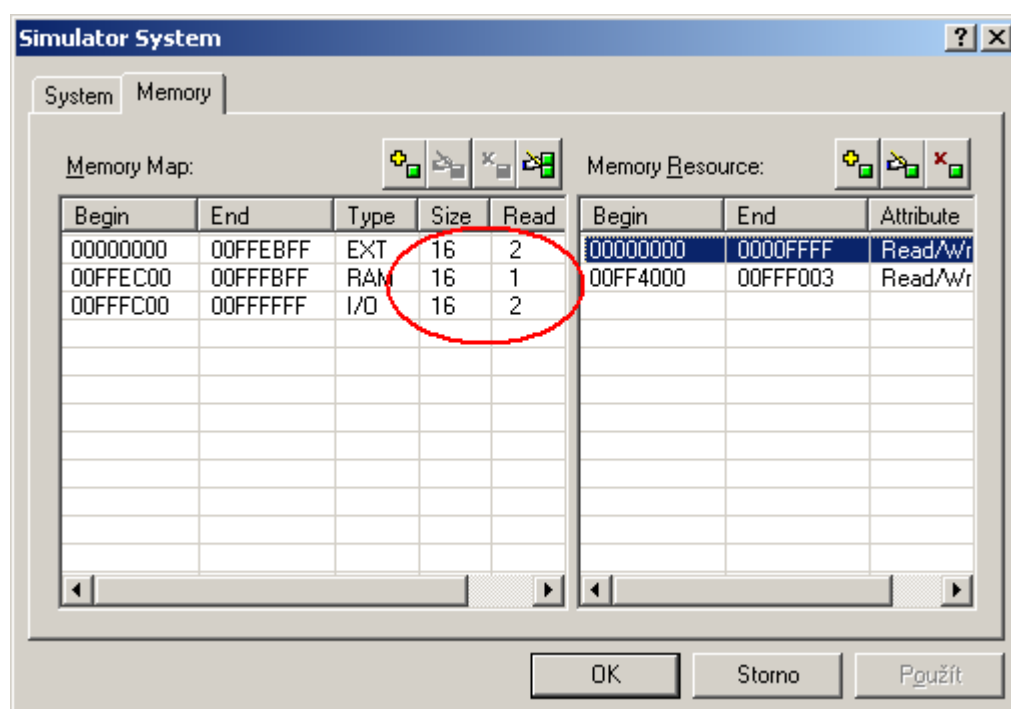
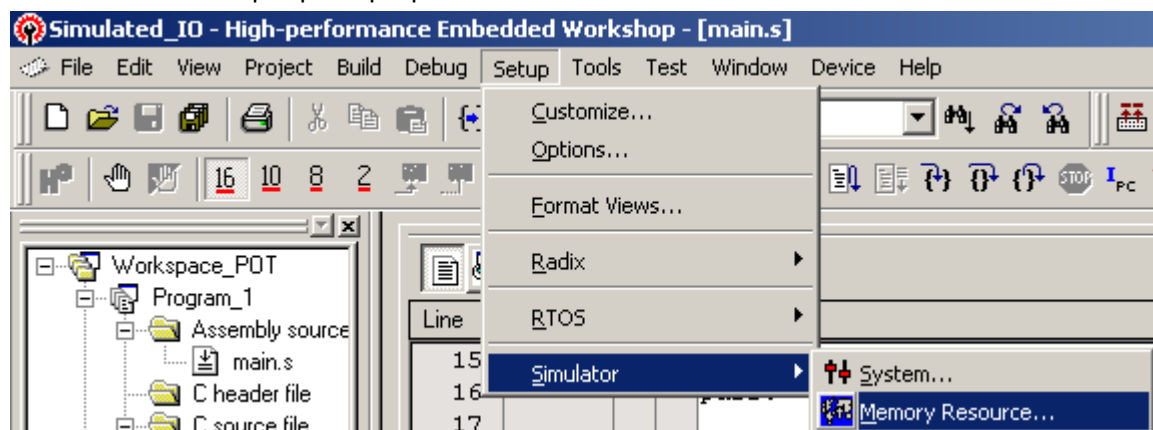
Zapnutí „performance profilu“



Okno, kde vidíme, kolik jednotlivé podprogramy potřebovaly taktů k vykonání programu.

Function/Variable	F/V	Address	Size	Times	Cycle	Ext mem	I/O area	Int mem
_start	F	000400	H'00000000	1	16	1	0	0

Nastavení vlastností pro přístup k paměti



Příklad:

V daném programu spočítejte, kolik taktů bude trvat, pokud jsou zapotřebí ke čtení z paměti 3 hodinové takty a sběrnice je 16 bitová.

```
mov.l #stck,ER7
push.w R0
push.w R1
mov.w @var1,R0
mov.w @var2,R1
add.w R1,R0
mov.w R0,@var3
```

lab1: jmp @lab1

Doby trvání jednotlivých instrukcí najdeme v programátorském manuálu:

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		
MOV.L #xx:32,ERd	R:W 2nd	R:W 3rd	R:W NEXT		
MOV.L ERs,ERd	R:W NEXT				
MOV.L @ERs,ERd	R:W 2nd	R:W:M NEXT	R:W:M EA	R:W EA+2	
MOV.L @(d:16,ERs),ERd	R:W 2nd	R:W:M 3rd	R:W NEXT	R:W:M EA	R:W EA+3

1. přístup k paměti (kód instrukce)

2. přístup k paměti (1. část konstanty)

3. přístup k paměti (2. část konstanty)

zpracování kódu + čtení další instrukce

			přístupů do paměti		vnitřní operace (při které se nenačítá další instrukce)
			čtení	zápis	
	mov.l	#stck,ER7	3x3		
	push.w	R0	1x3	1x3	1
	push.w	R1	1x3	1x3	1
	mov.w	@var1,R0	4x3		
	mov.w	@var2,R1	4x3		
	add.w	R1,R0	1x3		
	mov.w	R0,@var3	3x3	1x3	
lab1:	jmp	@lab1	2x3		1

Program bude trvat 69 taktů.