

Instrukcje PUSH i PULL – implementacja w mikrokodzie

Zaimplementować instrukcje operujące na stosie:

- PUSH – zapisanie (odłożenie) słowa na stosie
- PULL – odczytanie (zdjęcie) słowa ze stosu

Pliki znajdują się w katalogu lab3.

- wczytać konfigurację: *lab3.ecf*

- wczytać projekt: *lab3.mpr*

Stos (LIFO) ma posiadać następującą organizację:

- wskaźnik stosu (SP – Stack Pointer) wskazuje element na wierzchołku stosu (top)
- stos rośnie w kierunku malejących adresów
- wskaźnik stosu jest inicjowany wartością 0x400
- wskaźnik stosu jest dowolnym rejestrem R1 – R31

①

PUSH Rx, Ry

zapisz słowo z Ry na stos wskazywany przez Rx

1) $Rx - 4 \rightarrow Rx$

2) $Ry \rightarrow Mem[Rx]$

$Rx \rightarrow MAR$

$Ry \rightarrow MDR$

write memory (WW, MAR)

②

PULL Rx, Ry

odczytaj słowo ze stosu Rx do rejestru Ry

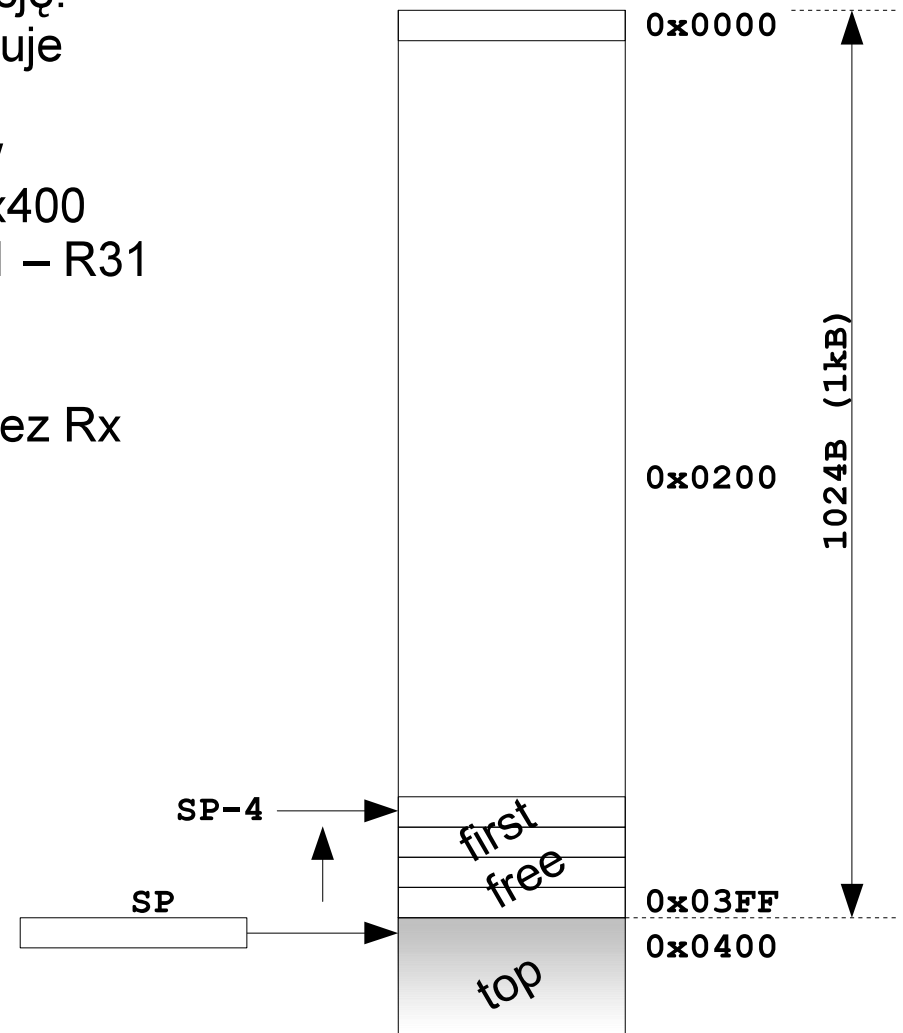
1) $Mem[Rx] \rightarrow Ry$

$Rx \rightarrow MAR$

read memory (RW, MAR, MDR)

$MDR \rightarrow Ry$

2) $Rx + 4 \rightarrow Rx$



Testowanie instrukcji PUSH – PULL

Sprawdzić (clock-by-clock) działanie instrukcji PUSH/PULL za pomocą programu:

```
ADDI R0, 0x????, R7
ADDI R0, 0xABCD, R1
PUSH R7,R1
PULL R7,R2
```

- 3 Wykonać program wykorzystujący instrukcje PUSH/PULL według przykładu.
Zmierzyć czas działania programu i sprawdzić wynik w R2 (0x2040)

0000:	44070400		ADDI R0, 0x????, R7	inicjowanie wskaźnika stosu R7
0004:	44010080		ADDI R0, 0x0080, R1	odłożenie na stos 128 słów o wartościach od 0x0080 do 0x0001
0008:	88E10000	next	PUSH R7,R1	
000C:	48210001		SUBI R1, 0x0001, R1	
0010:	7801FFFF		BRGT R1, next	
0014:	00000000		NOP	zdjęcie ze stosu 128 słów i sumowanie ich w rejestrze R2
0018:	34421000		XOR R2, R2, R2	
001C:	44010080		ADDI R0, 0x0080, R1	
0020:	8CE30000	next2	PULL R7,R3	
0024:	1C621000		ADD R3, R2, R2	
0028:	48210001		SUBI R1, 0x0001, R1	
002C:	7801FFFF		BRGT R1, next2	
0030:	00000000		NOP	
0034:	7000FFFC	halt	BRZ R0, halt	