

# Techniki mikroprocesorowe, laboratorium 1

Autorzy: Mateusz Koroś, Ksawery Pasikowski, Kamil Gabryjelski

## Zad. 1

Zadanie polegało na pokazaniu na oscyloskopie przebiegów cykli M1, memory write, memory read, IO read oraz IO write. W tym celu sygnały wyjściowe mikroprocesora Z80: RD, WR, IORQ i MREQ zostały podłączone do oscyloskopu. Kod, który wykorzystaliśmy do zaobserwowania owych cykli przedstawia się następująco:

```
# File zad1.lab1.asm
0000          ORG 1800h
1800          START:
1800 32 50 18          LD (1850h), A
1803 d3 00          OUT (00), A
1805 db 00          IN A, (00)
1807 18 f7          JR START
# End of file zad1.lab1.asm
1809
```

Procesor zaczyna pracę od komórki w pamięci o adresie 1800, pod którym znajduje się kod operacji LD, następnie pobiera z kolejnych dwóch komórek dwa bajty, przedstawiające adres 1850. W instrukcji tej widoczne są cykle M1, memory read i memory write. Następną instrukcją jest instrukcja OUT, której kod jest pobierany z komórki pamięci o adresie 1803. Z kolejnej komórki (o numerze 1804) pobierana jest liczba 00. W tej instrukcji obserwujemy cykle M1, memory read oraz IO write. Następnie z komórki 1805 mikroprocesor pobiera kod operacji IN a z kolejnej komórki 1806 liczbę 00. W instrukcji tej obserwujemy cykle M1, memory read i IO read. Ostatnią instrukcją, pobieraną z pamięci (komórka 1807) przez mikroprocesor jest instrukcja JR o kodzie operacji 18 a z kolejnej komórki pobierana jest liczba f7 (w kodzie U2), czyli -9 w systemie dziesiętnym. Następnie do rejestru rozkazów PC ładowana jest wartość 1800 (1809 – 9) i mikroprocesor wykonuje program od początku.

Poniżej znajdują się zdjęcia ekranu oscyloskopu, które przedstawiają poszczególne cykle.

### M1 oraz memory read



## Memory write



## IO write (OUT)

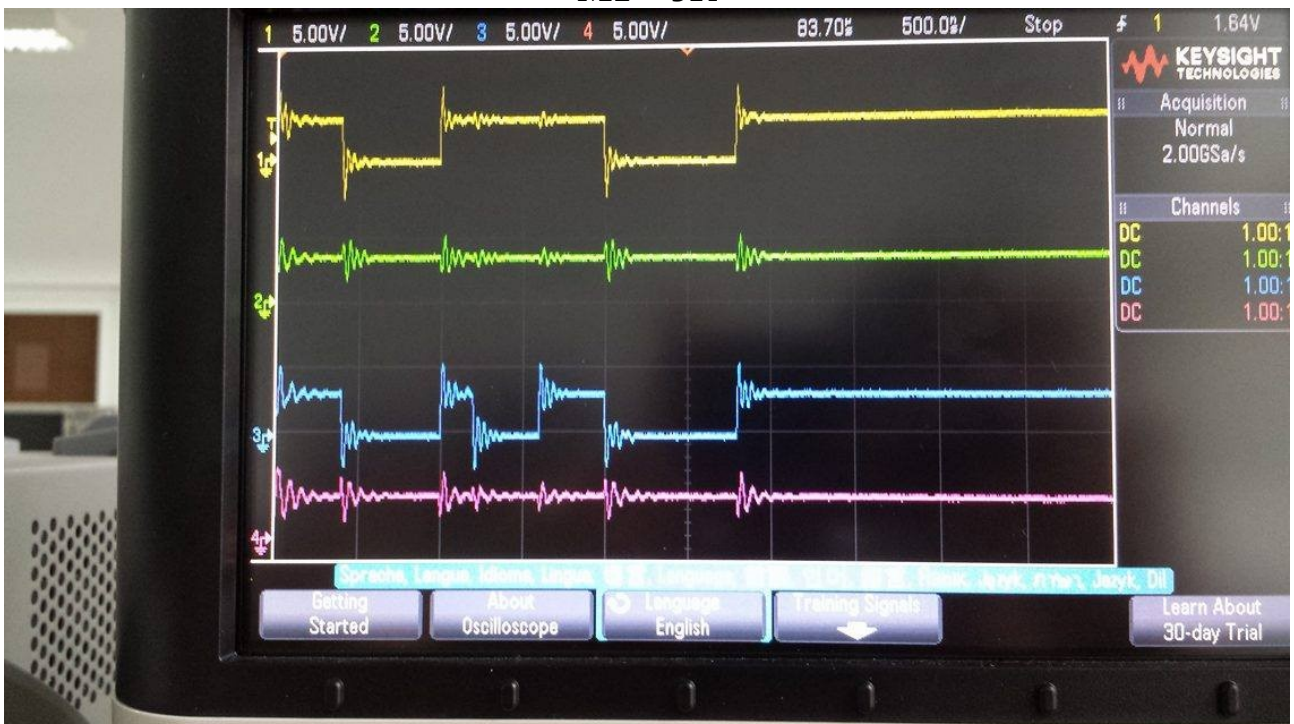




## IO read (IN)



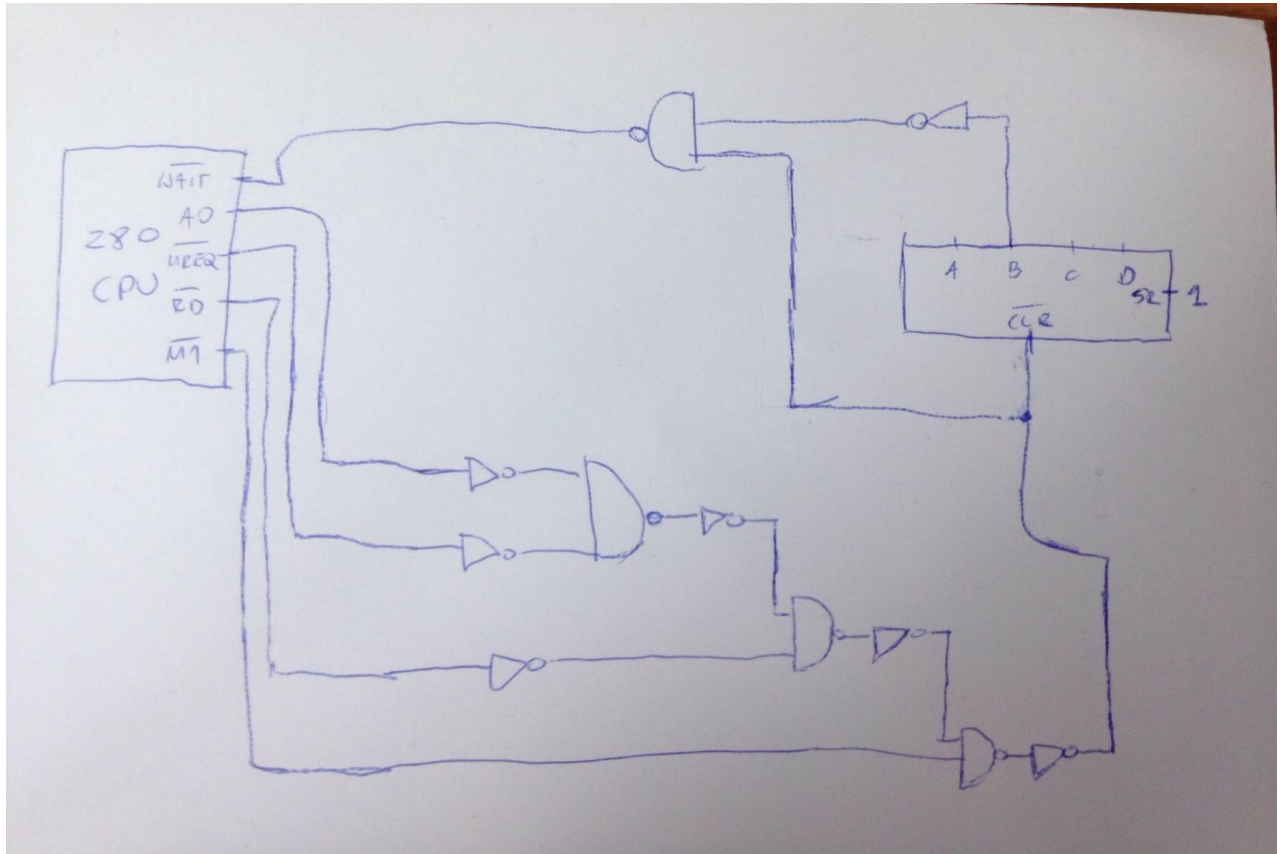
## M1 + JR



## Zad. 2

Zadanie polegało na zbudowaniu układu, który wstawia dwa dodatkowe takty zegara w cyklach odczytu z pamięci (memory read) przy parzystych adresach.

SCHEMAT UKŁADU :



Układ działa następująco:

1. Z bramek NAND dwunożnych zbudowana została bramka NAND czteronożna. Na wejścia tych bramek podłączone zostały sygnały -A0, -MREQ, -RD oraz -M1 (-M1 dlatego, że pierwszego cyklu każdej instrukcji nie traktujemy jako odczyt z pamięci).
2. Zanegowane wyjście czteronożnej bramki NAND zostało podłączone do wejścia -CLR rejestru przesunowego. W momencie gdy na -CLR pojawi się jedynka rejestr nie jest czyszczony i zaczyna się przesuwanie rejestru w prawo ( $S_0 = 0$ ,  $S_1 = 1$ ).  $S_R = 1$  powoduje, że na wejściu danych z prawej strony rejestru zawsze jest jedynka logiczna.
3. Zanegowane wyjście B rejestru przesunowego oraz zanegowane wyjście czteronożnej bramki NAND zostało podłączone do dwunożnej bramki NAND, której wyjście bezpośrednio połączone jest z wejściem -WAIT mikroprocesora Z80. Tak długo jak na B występuje zero logiczne (dwa takty zegarowe), tak wejście -WAIT pozostaje aktywny. W momencie, gdy na B pojawi się jedynka logiczna, sygnał -WAIT przestaje być aktywny.

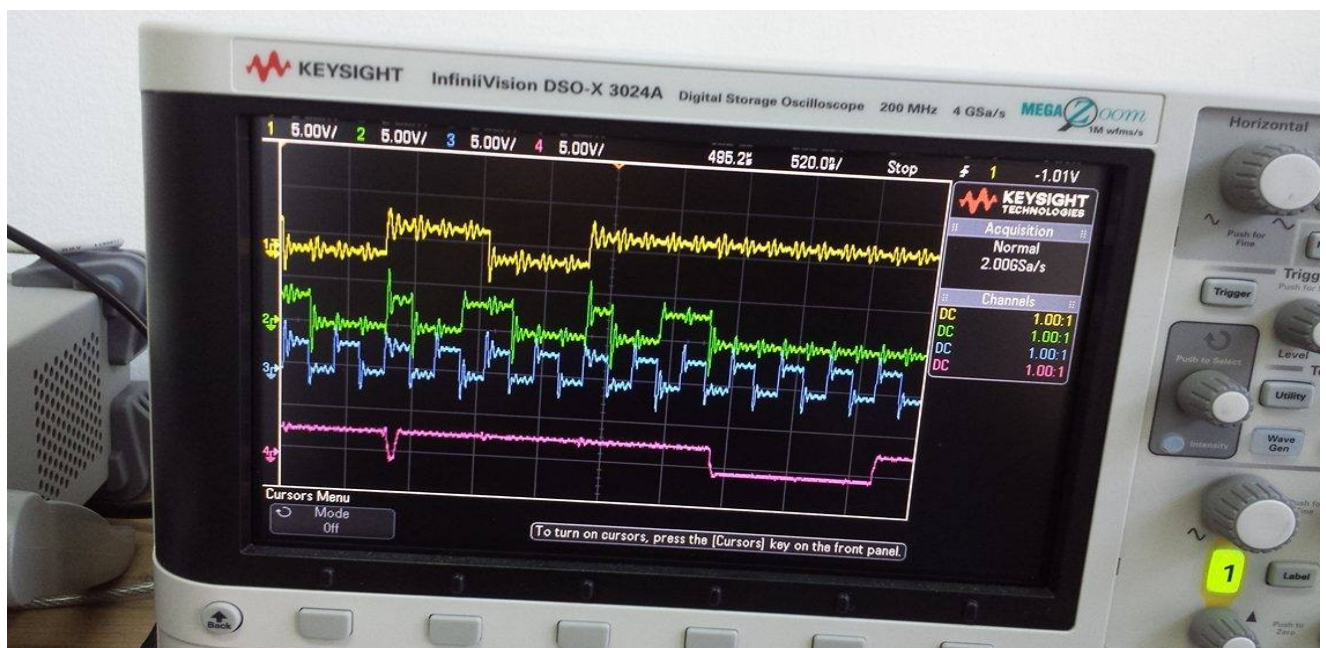


Kod, który wykorzystaliśmy do zaobserwowania na oscyloskopie działania naszego układu przedstawia się następująco :

0000	ORG 1800h
1800	START:
1800 00	NOP
1801 32 50 18	LD (1850h), A
1804 00	NOP
1805 d3 00	OUT (00), A
1807 00	NOP
1808 db 00	IN A, (00)
1810 18 f4	JR START

Rozkazy NOP zostały wstawione do programu, aby przesunąć odpowiednie bajty w pamięci, tak aby przetestować odczyt z pamięci zarówno z komórek parzystych jak i nieparzystych.

Poniżej znajduje się zdjęcie ekranu oscyloskopu, na których widać, że w trakcie odczytu z pamięci układ wstawia dodatkowe dwa takty zegarowe.



Legenda :

sygnał żółty	- M1
sygnał zielony	- MREQ
sygnał niebieski	- CLOCK
sygnał fioletowy	- WAIT