**TECHNIKA MIKROPROCESOROWA**

**Sprawozdanie, lab. Nr. 1**

Paweł Walczak

Konrad Sobolewski

Antek Różański

1. **Zadanie nr. 1**

**Treść**

Opracować i zaprezentować program generujący cykle: *M* 1, *MR*, *MW* ,

*IOR*, *IOW* .

## Wykonanie:

Wygenerowanie wszystkich zadanych cykli wymaga wgania i uruchomienia na procesorze programu składającego się z konkretnych instrukcji. Poniżej wymieniamy cykle, które chcemy zaobserwować wraz z ich opisem oraz intruskcje, podczas których można je zaobserwować na oscyloskopie:

* + 1. *M*1 - cykl pobrania polecenia. Pojawia się w każdym cyklu instrukcji, ponieważ kod każdego polecenia musi być pobrany przed swoim wykonianiem. Cykl łatwy do rozpoznania dzięki sygnałowi *M*1, który sygnalizuje pobyt procesora w owym cyklu wraz z towarzyszącymi mu dwoma niewielkimi skokami na linii MREQ.
    2. *MR* - cykl odczytu z pamięci. Uzyskaliśmy go przez istrukcję LD (polecenie to pobiera dane z adresu podawanego jako drugi argument i umieszcza go w rejestrze lub wskazanym przez pierwszy argument). Polecenuie LD wywołuje również cykle *M*1 i MW. Na oscyloskopie objawia się przez aktywne równocześnie sygnały *MREQ* i *RD*.
    3. *MW* - cykl zapisu do pamięci. Wygenerowalismy poleceniem opisanym powyżej. Rozpoznać go można po aktywnych równocześnie sygnałach *MREQ* i *WR*.
    4. *IOR* - cykl odczytu z portu urządzeń peryferyjnych. Wymusza go intrukcja *IN* , która służy do odczytu wartości z portów zewnętrznych i zapisu ich do danego miejsca w pamięci. Cykl można rozpoznać poprzez równocześnie aktywne sygnały *IORQ* i *RD*.
    5. *IOW* - cykl zapisu do portu urządzeń peryferyjnych,generowany podczas polecenia *OUT* , które zapisuje wartość z danego miejsca w pamięci do urządzenia zewnętrznego. Cykl można rozpoznać poprzez równocześnie aktywne sygnały *IORQ* i *WR*.

Dane wygenerowane po skompilowaniu I uruchomieniu programu:

|  |  |  |
| --- | --- | --- |
| 1 | 0000 | ORG 1800h |
|  | 1800 | START: |
|  | 1800 32 01 20 | LD (0x2001), A |
|  | 1803 db 01 | IN A, (0x01) |
|  | 1805 d3 02 | OUT (0x02), A |
|  | 1807 18 f7 | JR START |
|  |  |  |

Cały program został zawarty w prostej pętli, która wykonuje się w nieskonczoność. Jak można zauważyć na oscyloskopie funkcja skoku (JR) też wywołuje szereg cykli.

Na ostatniej stronie dołączamy przebiegi czterech sygnałów (MREQ, IORQ, RD, WR) pobrane z ekranu oscyloskpu, na których widać wszystkie podstawowe cykle.

# Zadanie nr. 2

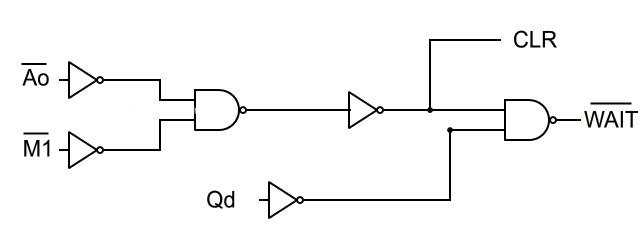
## Treść:

Zaprojektować układ, który dla cykli M1 o parzystych adresach wstawia dodatkowo 3 cykle WAIT z możliwością łatwej przebudowy układu w celu wstawienia innej liczby dodatkowych cykli WAIT.

## Wykonanie:

Aby zrealizować zadanie należy wykonać szereg operacji. Po pierwsze należy stworzyć układ, który będzie wykrywał interesujący nas cykl. W naszym przypadku sygnał oraz *A0* muszą być w stanie niskim. Odliczanie liczby taktów odbywa się za pomocą modułu rejestru przesuwnego. Gdy zostanie wykryty interesujący nas cykl, ustawiamy wejście CLR rejestru w stan wysoki (rejestr przestaje zerować swój stan). Z każdym cyklem zegara rejestr wówczas przesuwa wartość w prawo. Sygnał WAIT jest wstawiany do momentu wykrycia jedynki na określonej pozycji rejestru (dla 3 ykli wstawiania jest to pozycja D). Po wykryciu owej jedynki rejestr jest zerowany za pomocą sygnału CLR ustawianego na stan niski.

Stworzony przez nas układ kombinacyjny korzysta z dostępnych w laboratorium bramek NAND oraz negatorów. Korzystamy również w wyżej wspomnianego rejestru przesuwnego oraz płytki z przełącznikami do korzystania ze stałych wartości logicznych 0 i 1.



Rysunek 1: Schemat połączenia układu wyzwalania sygnałów *WAIT* .

*CLK* 1



*A B C D*

*CLK CLR*

R1

*SR A B C D*

*S*0

*S*1 *SL*

*A B C D*

*CLK CLR*

R2

*SR A B C D*

*S*0

*S*1 *SL*

*Q*7 *Q*6 *Q*5 *Q*4

*Q*3

*Q*2 *Q*1 *Q*0

1 *CLK*

0 *F*

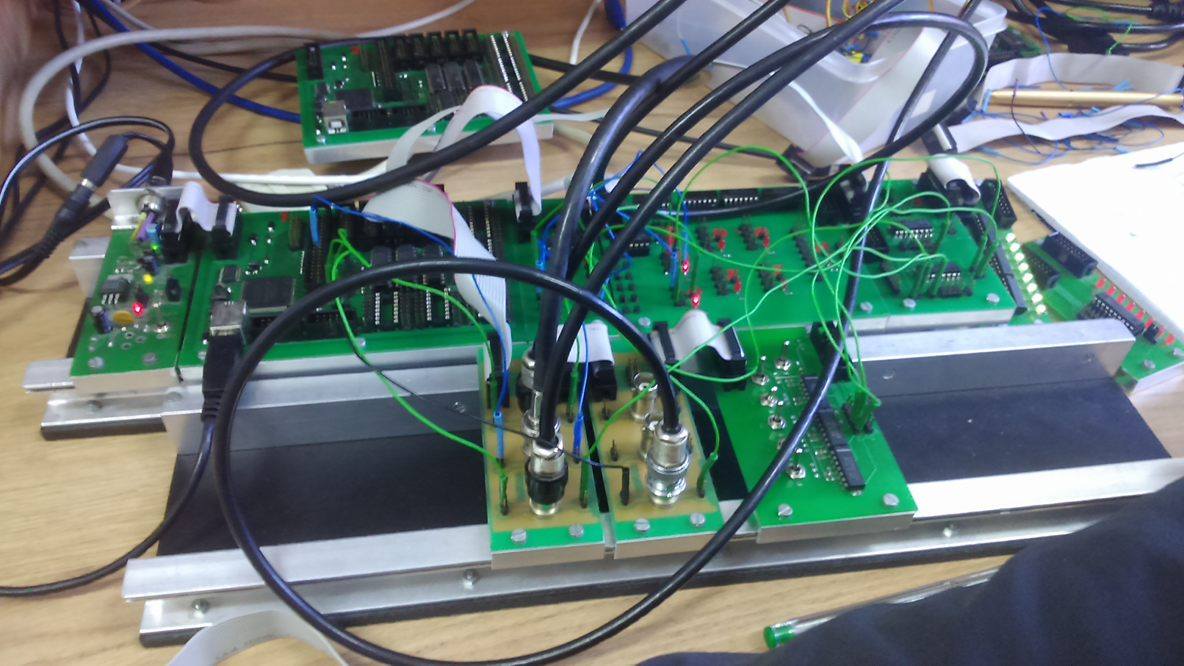
*F* 0

1

Rysunek 2:

Sposób połączenia dwóch rejestrów czterobitowych.

*F* - funkcja wykrywająca zadany cykl.



Rysunek 3. Schemat połączenia układu w laboratorium.

## Test i wyświetlanie przebiegu na oscyloskopie

Do testowania tak działającego układu użyliśmy programu z zadania 1.

Listing 2: Program do testowania układu wstawiania taktów oczekiwania.

|  |  |  |
| --- | --- | --- |
| 1 | 0000 | ORG 1800h |
|  | 1800 | START: |
|  | 1800 db 18 | IN A, (18h) |
|  | 1802 32 20 18 | LD (1820h), A |
|  | 1805 d3 18 | OUT (18h), A |
|  | 1807 18 f7 | JR START |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |

Na ostatniej stronie zapisany jest przebieg czterech istotnych dla zadania sygnałów, odczytany z oscyloskopu.

# Zadanie nr 3

## Treść:

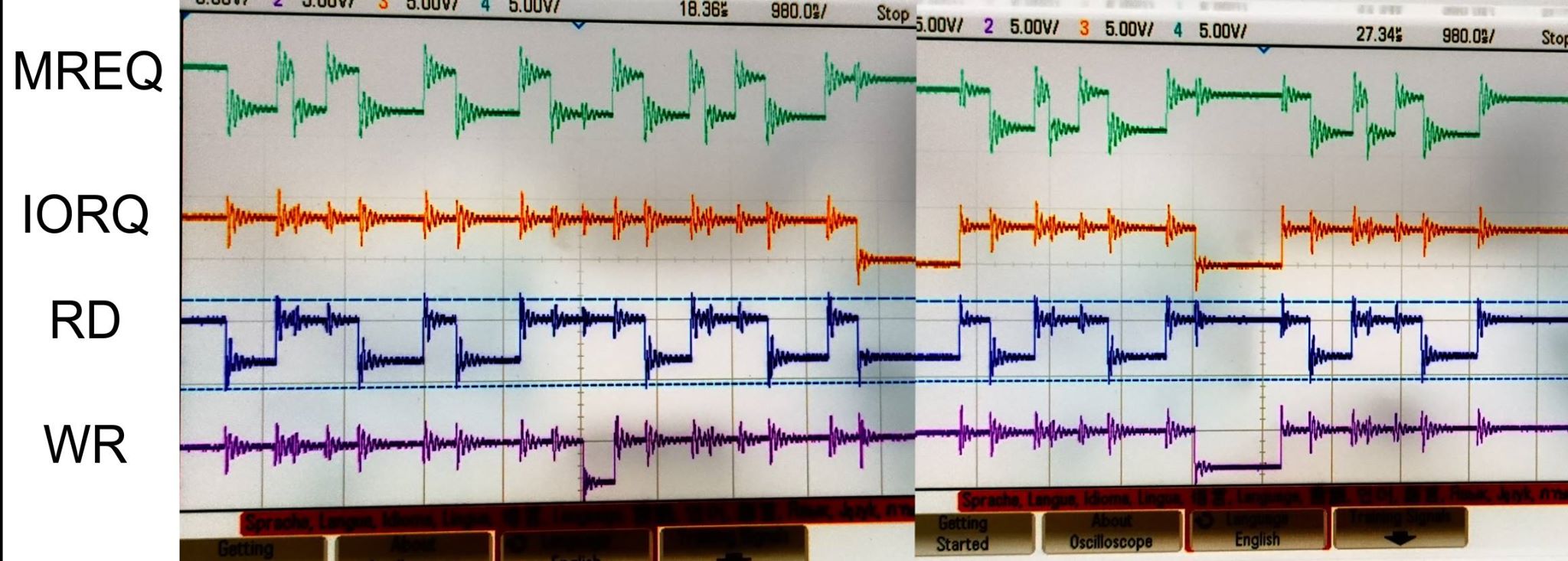
Opracować i zaprezentować układ zatrzymywania procesora w cyklach opi- sanych w zadaniu 2. tak, aby takty były wstawiane do czasu akcji użytkow- nika.

## Wykonanie:

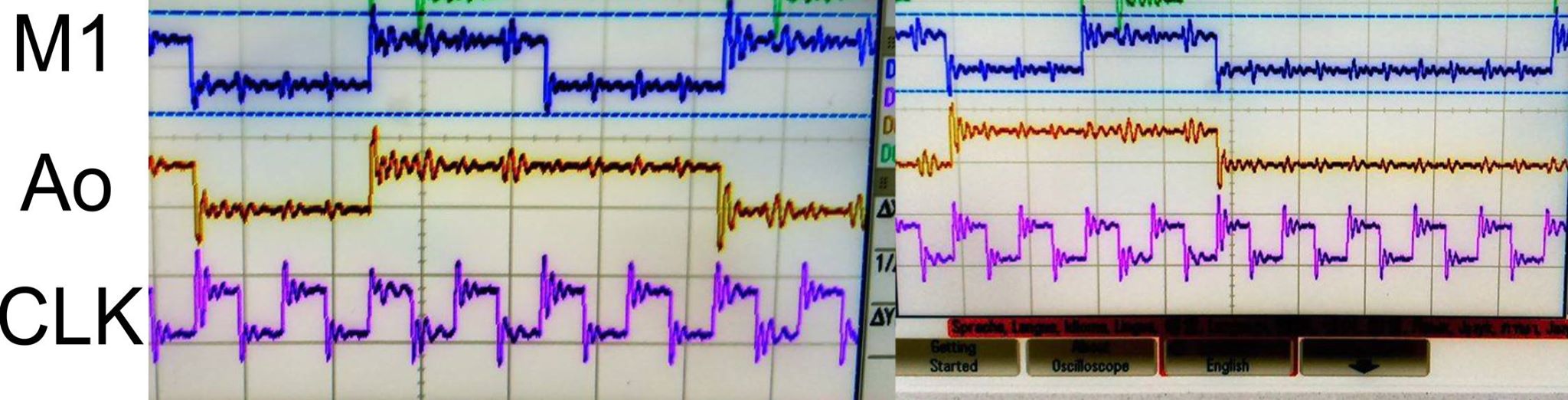
Układ ten jest jedynie lekką modyfikacją układu z zadania drugiego. W tam tym układzie rejestr przesuwny używany był do generacji sygnału przerwania wstawiania taktów i działał on zgodnie z zegarem procesora. Pomysłem na modyfikację tego układu jest rezygnacja ze sprzężenia tych dwóch zegarów. Zamiast tego zegar rejestru połączyć z przełącznikiem, a jako bit przery- wania procesu wstawiania taktów oczekiwań ustawić pierwszy bit rejestru. Układ wyjdzie ze stanu oczekiwania dopiero po takcie zegara rejestru, a ten będzie zależał od akcji użytkownika.

Z powodu braku czasu w labolatorium układ ten nie został przetestowany, więc jego działanie nie jest w żaden sposób udokumentowane

5



Rysunek 4: Przebieg sygnałów dla zadania pierwszego



Rysunek 4: Przebieg sygnałów dla zadania drugiego(lewa strona zdjęcia, cykl bez wydłuzenia, prawa strona, cykl z wydłuzeniem.