

**Wydział Fizyki, Astronomii i Informatyki Stosowanej**  
**Uniwersytetu Mikołaja Kopernika**  
**Pracownia układów programowalnych**

**Ćwiczenie 11**

**Wykorzystanie bloków pamięciowych RAMB w układach FPGA**  
**(RAMB, XC2V1000, V2MB1000, RAMB)**

### Cel ćwiczenia

Celem ćwiczenia jest zapoznanie studenta z możliwościami bloków pamięciowych RAMB będących obecnie jednym z podstawowych zasobów układu programowalnego FPGA. W trakcie zajęć student zdobędzie praktyczną umiejętność konfiguracji, implementacji i wykorzystania w. w. bloków pamięciowych poprzez opis w języku VHDL.

### Zagadnienia do przygotowania

**Przed** przystąpieniem do ćwiczenia student powinien być zapoznany z działaniem bloku pamięci RAMB16\_Sm\_Sn [1] skonfigurowanego jako RAMB16\_S9\_S9. Plik `main.vhd` powinien zostać przeanalizowany tak, aby sposób wykorzystania bloku pamięciowego RAMB nie budził **żadnych** wątpliwości. W szczególności należy przyjrzeć się procesowi PROC\_READ oraz konstrukcji mapowania bloku pamięciowego.

Mapowania `db1`, `db2` implementują moduły eliminacji wpływu drgań styków przycisków SW5 i SW6 na stan układu, natomiast mapowania `dc1` oraz `dc2` implementują konwertery kodu (kod binarny naturalny na kod 7 segmentowy). Nie jest wymagana ich dogłębna analiza.

Dodatkowo wymaga się przygotowania (**przed zajęciami**) algorytmu oraz diagramu czasowego procesu zapisu do pamięci oraz zarysu kodu w języku VHDL, który taki zapis będzie przeprowadzał – należy wzorować się na procesie PROC\_READ.

### Przebieg ćwiczenia

1. Wykorzystując pliki źródłowe (`main.vhd`, `main.ucf`, `debouncer.vhd` oraz `decoder.vhd`) utworzyć nowy projekt i dokonać jego implementacji do struktury FPGA. **Wybrać strukturę {Virtex2, XC2V1000, FG456, -4}.**

Odczytać zawartość pamięci RAMB. Każde naciśnięcie przycisku SW6 powoduje przejście do kolejnego etapu cyklu odczytu pamięci. Pierwsze naciśnięcie SW6 rozpoczyna cykl odczytu – wówczas zespołem przycisków SW4 możliwe jest ustalenie 8 najmłodszych bitów adresu. Drugie wciśnięcie SW6 powoduje odczytanie adresu z SW4 i podanie go na magistralę adresową bloku RAMB – wówczas wyświetlacz jest wygaszony. Po trzecim wciśnięciu przycisku SW6 na wyświetlaczu pojawi się zawartość komórki pamięci, której mniej znaczące 8 bitów adresu zostało pobrane 2 etapy wcześniej z zespołu przełączników SW4. W celu zakończenia cyklu odczytu należy dokonać jeszcze dwóch dodatkowych wciśnień przycisku SW6. Podczas całego cyklu zapisu dioda USER jest *zapalona*.

Odczytać wszystkie te komórki pamięci, które zostały zainicjalizowane niezerową wartością w pliku `main.vhd` oraz sprawdzić czy pozostałe komórki pamięci mają wartość równą 0x00 (wybrać maksymalnie 5 różnych adresów).

2. Wzorując się na procesie PROC\_READ napisać proces PROC\_WRITE, który będzie

- umożliwił zapis do komórki pamięci RAMB wskazanej zespołem przełączników SW4 w jednym cyklu wartości pobranej z zespołu przełączników SW4 w kolejnym cyklu. Kolejne etapy zapisu powinny zostać wybierane przyciskiem SW5. Dioda USER powinna być zapalona przez cały cykl zapisu – analogicznie jak w punkcie 1.
3. Dokonać modyfikacji wartości pięciu wybranych komórek pamięci (przycisk SW5). Sprawdzić (przycisk SW6) poprzez odczyt komórek zmodyfikowanych, poprawność wykonania punktu 2.
  4. (punkt obowiązkowy dla 60-cio godzinnych grup ćwiczeniowych) W procesie dokonującym odczytu pamięci zaimplementować możliwość wyboru 3 najstarszych bitów adresu. W tym celu dodać w procesie PROC\_READ cykle maszynowe, które pozwolą na pobranie tych 3 bitów z zespołu przełączników SW4 oraz przekażą ich stan na wejścia pamięci ADDRA poprzez sygnał aa. Aby sprawdzić działanie tej części należy pod adresy odległe co 256 bajtów wpisać słowa referencyjne (w konstrukcji `generic map`) i po implementacji do struktury FPGA dokonać ich odczytu.

## Literatura

[1] Libraries Guide – ISE 6.3. Xilinx. (plik lib.pdf)

[2] Virtex-II™ V2MB1000 Development Board User's Guide. Memec. 2002. (plik V2MB\_User\_Guide\_3\_0.pdf)

**UWAGA**, wszystkie pozycje literaturowe dostępne są w postaci elektronicznej (pliki PDF).