

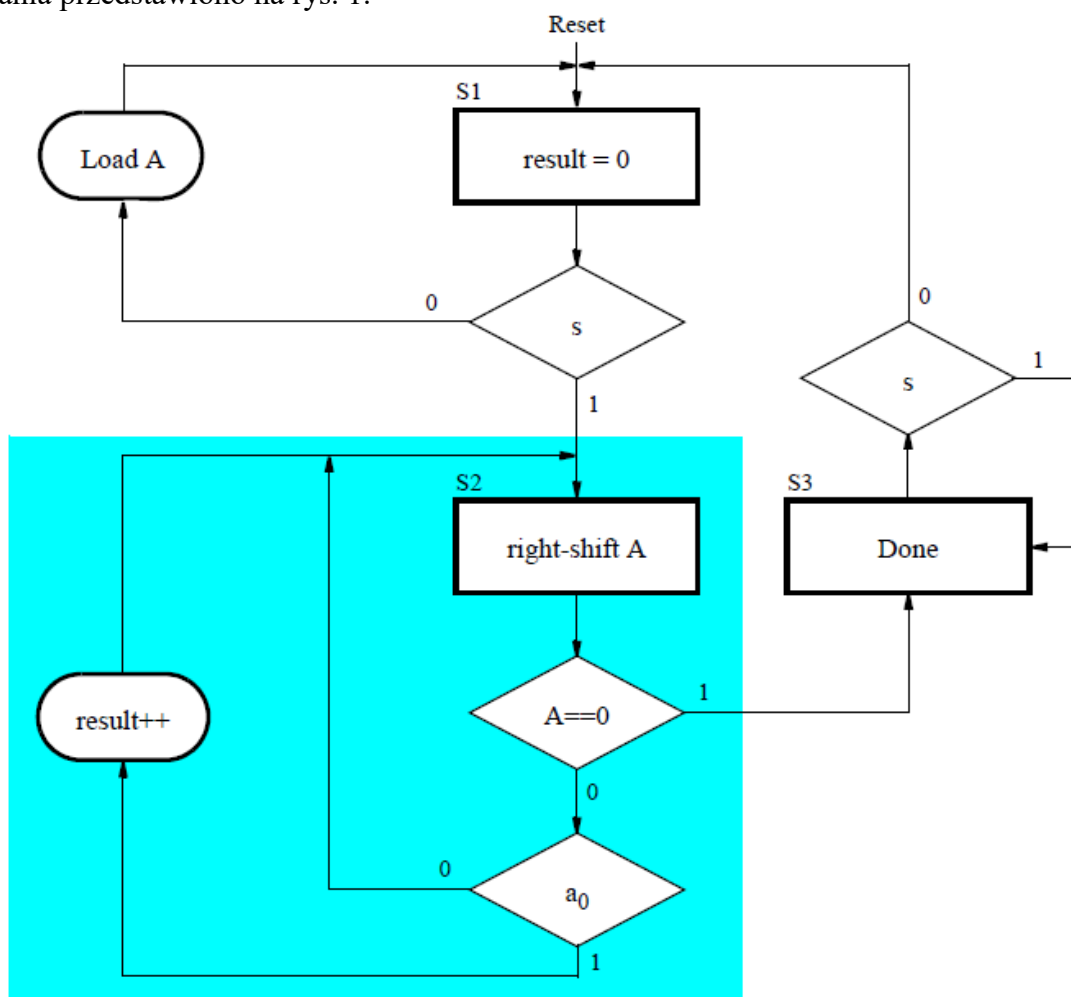
Lab. 12. Sprzętowa realizacja algorytmów

Wymagana wiedza

- Algorytmiczny automat skończony (**ASM**) (automat mikroprogramowany);
- diagramy **ASM** (diagramy przepływu danych – flow charts);
- niejawna synchronizacja (**implied timing**).

Wykonanie

1. Utwórz projekt liczba_bitów, który określa liczbę jedynek w 8-bitowym słowie A, którego algorytm działania przedstawiono na rys. 1.



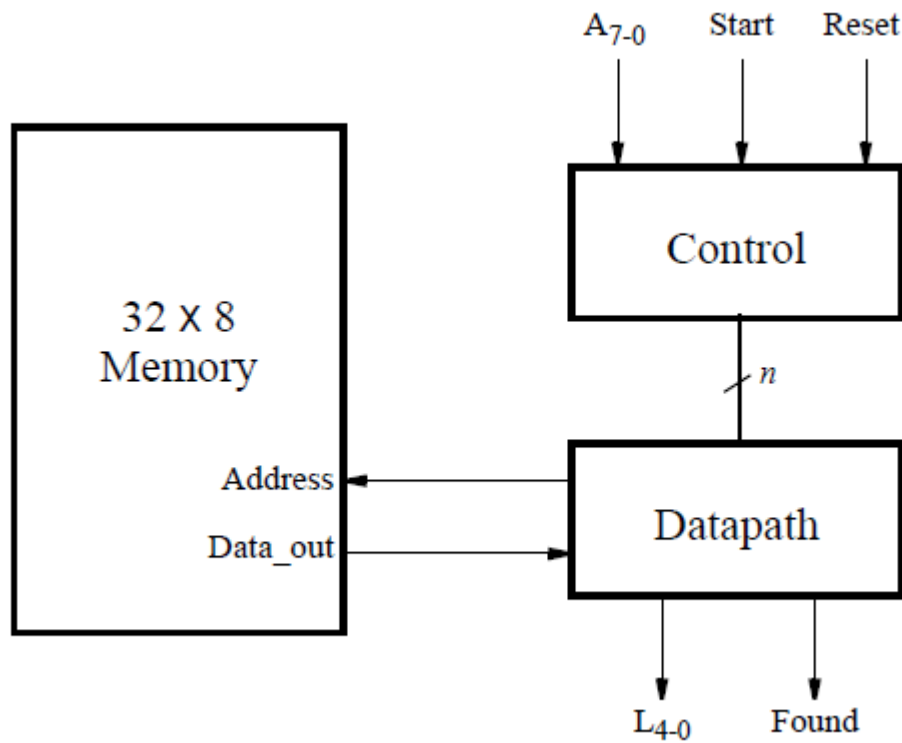
Rys. 1. Wykres ASM dla obwodu zliczającego liczbę jedynek w słowie A

Z następującym przyporządkowaniem pinów:

Wyprowadzenia	Piny
in	SW7-0 – wejścia układu
clk	CLOCK_50
sclr	KEY0
Run	SW9
result	HEX0
Done	LEDR9

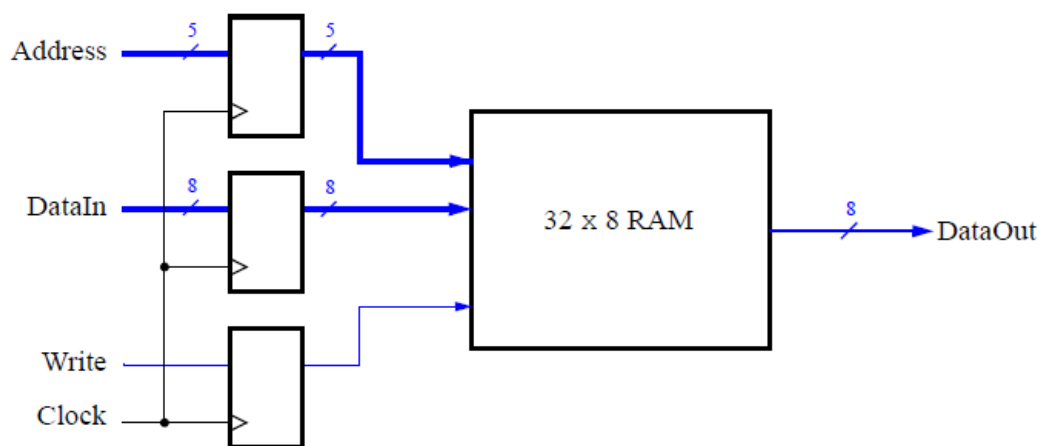
Wskazówka: Użyj jako A rejestru przesuwającego. Policz liczbę jedynek za pomocą 4-bitowego licznika.

2. Utwórz projekt `binary_search`, który implementuje binarny algorytm wyszukiwania w tablicy 8-bitowej wartości określonej przy pomocy przełączników SW7-0. Schemat blokowy projektu przedstawiono na rys. 2.



Rys. 2. Schemat blokowy układu binarnego poszukiwania

Tablica wartości znajduje się w module pamięci, którego schemat pokazano na rys. 3..



Rys. 3. Pamięć RAM 32x8 z rejestrem adresów

Przykład pliku **MIF** do inicjalizacji pamięci pokazano na rys. 4.

```

DEPTH = 32;
WIDTH = 8;
ADDRESS_RADIX = HEX;
DATA_RADIX = BIN;
CONTENT
BEGIN

00 : 01;
01 : 02;
02 : 03
03 : 05;
04 : 06;
05 : 06;
06 : 07;
.... (some lines not shown)
1E : 1F;
1F : 20;

END;

```

Rys. 4. Przykład pliku inicjującego pamięć (**MIF**)

Schemat projektu musi utworzyć 5-bitowe słowo wyjściowe **L**, które określa adres szukanej liczby **A**, a także wyjście **found**, które wynosi 1, jeśli liczba **A** znajduje się w tablicy, oraz 0, jeśli liczba **A** nie znajduje się w tablicy. Projekt powinien mieć następujące przypisania pinów:

Wyprowadzenia	Piny
Start	SW9
A	SW7-0
Resetn	KEY0
clk	CLOCK_50
L	HEX1, HEX0
found	LEDR9

Wykonaj symulację i przetestuj projekt na płycie.