

Paweł Borowski

Rafał Malinowski

Cezary Zawadka

SNAKE

DOKUMENTACJA WSTĘPNA

Projekt z przedmiotu Układy Cyfrowe

Prowadzący: mgr inż. Michał Staworko

1. Informacje ogólne:

- Zarys historyczny:

Gry typu „Snake” zostały zapoczątkowane przez grę pt. „Blockade”, wydaną w 1976 r. przez firmę Gremlin. Każdy gracz sterował w niej wężem, za którego pomocą musiał zablokować przeciwnika tak, aby ten wpadł na przeszkodę. Większa popularność zyskał jednak „Snake” w znanej nam formie, co poskutkowało jego wydaniem na wielu platformach. Swoje odrodzenie gra przeżyła w 1998 r. kiedy została dodana jako standardowa aplikacja do komórek firmy Nokia.

- Zasady gry:

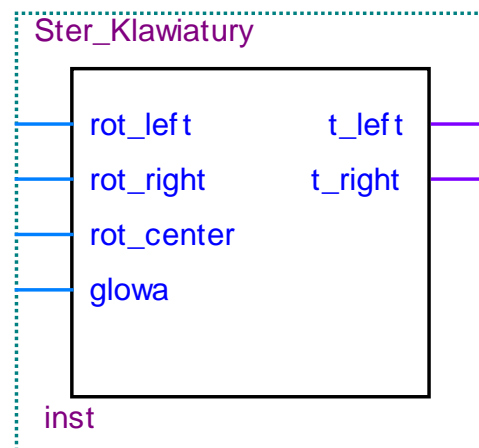
Gra odbywa się na planszy o rozmiarach 20 kolumn na 20 wierszy, po której porusza się tytułowy wąż. Zadaniem gracza jest takie sterowanie wężem by ten nie wpadł na przeszkody (krawędź planszy lub własny ogon) i jednocześnie zbieranie pojawiających się na planszy bonusów. Każdy zebrany bonus powoduje wydłużenie węża oraz dodanie 1 pkt do wyniku. Gracz nie ma możliwości zatrzymania węża, może jedynie zmienić kierunek ruchu o 90 stopni.

2. Proponowany sposób realizacji:

- W pamięci RAM będziemy przechowywać położenie węża oraz położenie bonusu.
- Rozmiar planszy 20x20 makropikseli, z czego każdy ma rozmiar 16x16 pikseli.
- Prosty system punktacji: 1 bonus = 1 punkt.
- Sterowanie za pomocą pokrętła będącego częścią płytki Spartan-3e.
- Gra będzie wyświetlana na ekranie monitora VGA.

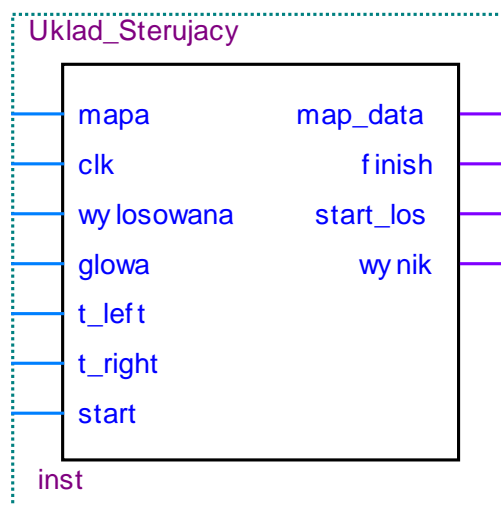
3. Opis funkcji poszczególnych elementów i sposób ich współdziałania:

3.1 Układ sterujący klawiaturą:



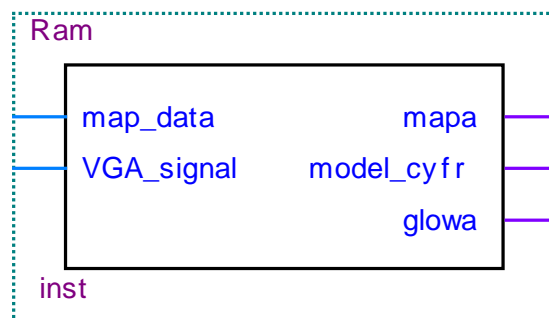
Wejścia	Wyjścia
rot_left[bit] – stan wysoki gdy pokrętko w pozycji „left”	t_left[bit] – stan wysoki gdy wąż ma skręcić w lewo
rot_right[bit] – stan wysoki gdy pokrętko w pozycji „right”	t_right[bit] – stan wysoki gdy wąż ma skręcić w prawo
rot_center[bit] – stan wysoki gdy pokrętko w pozycji „right”	
Głowa[9..0] – informacja o pozycji głowy węża	

3.2 Układ sterujący grą:



Wejścia	Wyjścia
clk[bit] – stan wysoki gdy następuje tyknięcie zegara	map_data[10..0] – wysyła do pamięci RAM prośbę o zmianę sytuacji na planszy
Start[bit] – stan wysoki gdy następuje włączenie gry	Finisz[bit] – stan wysoki gdy następuje śmierć lub wygrana
Mapa[10..0] – info o zawartości wybranego pola	start_los[bit] – stan wysoki gdy wysyła prośbę o wygenerowanie pola do układu losującego położenie bonusu
Wylosowana[9..0] – informacja o pozycji bonusu	Wynik[bit] – stan wysoki gdy wysyła prośbę do układu VGA o doliczenie punktów
Głowa[9..0] – informacja o pozycji głowy węża	
t_left[bit] – stan wysoki gdy wąż ma skręcić w lewo	
t_right[bit] – stan wysoki gdy wąż ma skręcić w prawo	

3.3 RAM:



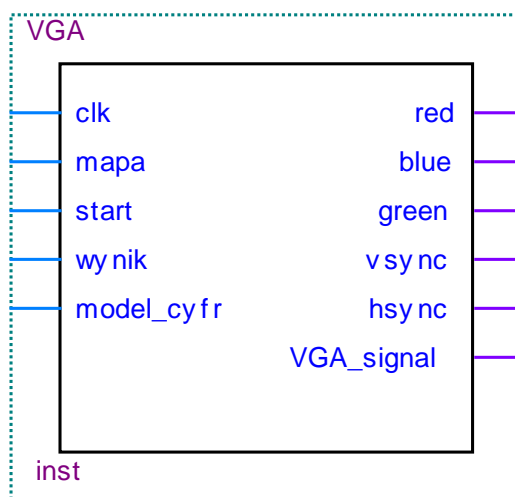
Wejścia	Wyjścia
map_data[10..0] – odbiera z układu sterującego prośbę o zmianę sytuacji na planszy	Mapa[10..0] – informacja o zawartości następnego pola na którym znajdzie się głowa węża
VGA_signal[9..0] – odbiera z układu VGA adres komórki w pamięci, której stan chciałby przestawić na ekranie układ VGA	Głowa[9..0] – informacja o pozycji głowy węża
	model_cyfr[7..0] – modele cyfr potrzebne do wyświetlania wyniku.

3.4 Układ losujący położenie bonusu:



Wejścia	Wyjścia
start_los[bit] – stan wysoki gdy następuje prośba o wylosowanie pozycji na której znajdzie się bonus	Wylosowana[9..0] – informacja o pozycji bonusu
clk[bit] – stan wysoki gdy następuje tyknięcie zegara	

3.5 Układ kontrolera VGA:



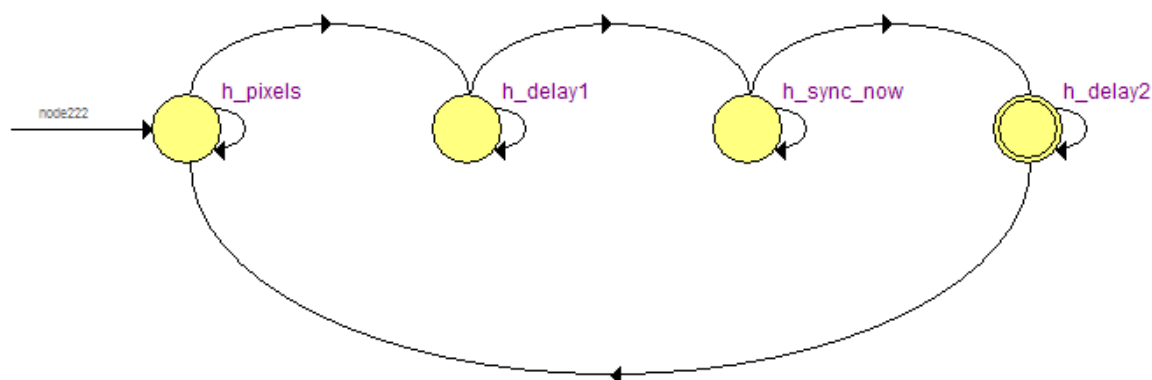
Wejścia	Wyjścia
Mapa[10..0] – info o zawartości wybranego pola	RGB[2..0] – sygnały składowych koloru, na ekranie można uzyskać 8 kolorów.
clk[bit] – stan wysoki gdy następuje tyknięcie zegara	Vsync[bit] – stan niski powoduje synchronizację pionową obrazu (początek

	rysowania nowej ramki)
Start[bit] – stan wysoki gdy następuje włączenie gry	Hsync[bit] - stan niski powoduje synchronizację poziomą obrazu (początek rysowania nowej linii)
Wynik[bit] – stan wysoki gdy nastąpi prośba o dopisanie punktów	VGA[9..0] – wysyła adres komórki w pamięci której stan układ VGA chciałby wypisać na ekran
mod_cyfr[7..0] - modele cyfr potrzebne do wyświetlania wyniku.	

4. Automaty synchroniczne kontrolera VGA.

Kontroler VGA będzie realizowany przez dwa automaty, *h_machine* oraz *v_machine*, które będą odpowiedzialne za synchronizację poziomą i pionową obrazu. Na ekranie wyświetlany będzie obraz o rozdzielczości 640x480, o odświeżaniu pionowym 60Hz.

4.1 Diagram Stanów automatu *h_machine* wygląda następująco:

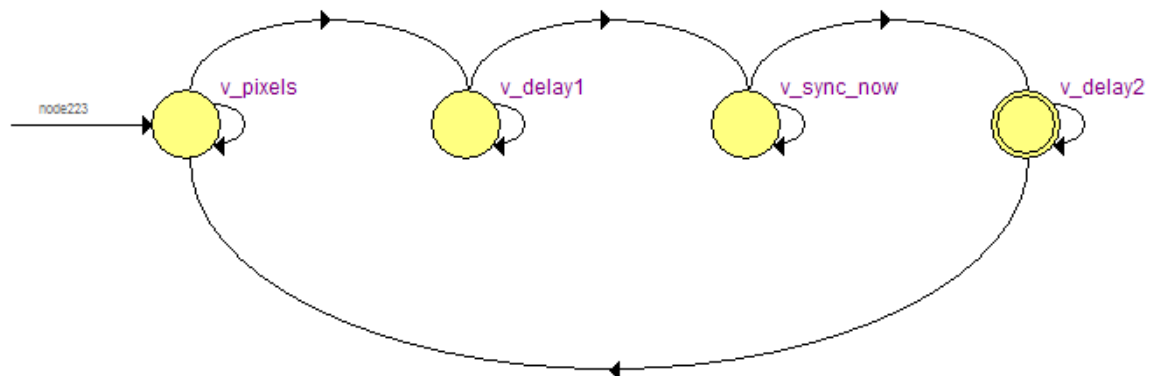


Tablica przejść automatu *h_machine*:

stan aktualny	stan następny	warunek przejścia
h_sync_now	h_delay2	Licznik_H!=1493
h_sync_now	h_sync_now	Licznik_H=1493
h_pixels	h_delay1	Licznik_H=1258
h_pixels	h_pixels	Licznik_H!=1258
h_delay2	h_delay2	Licznik_H!=1588
h_delay2	h_pixels	Licznik_H=1588
h_delay1	h_delay1	Licznik_H=1305
h_delay1	h_sync_now	Licznik_H!=1305

Płyta laboratoryjna Spartan-3E Starter Kit na której mamy zamiar zrealizować nasz układ jest taktowana zegarem 50MHz, dlatego o przejściu do następnych stanów decyduje *Licznik_H* liczący takty zegara modulo 1589 (co odpowiada 31.77 μ s, jest to czas rysowania jednej linii razem z czasami „Front porch”, „Sync pulse”, „Back porch”). W stanie *h_sync_now* na wyjściu *hsync* ustawiony jest stan niski.

4.2 Diagram Stanów automatu *v_machine* wygląda następująco:



Tablica przejść automatu *v_machine*:

stan aktualny	stan następny	warunek przejścia
v_pixels	v_pixels	Licznik_V!=479
v_pixels	v_delay1	Licznik_V=479
v_delay1	v_delay1	Licznik_V!=491
v_delay1	v_sync_now	Licznik_V=491
v_sync_now	v_sync_now	Licznik_V!=493
v_sync_now	v_delay2	Licznik_V=493
v_delay2	v_pixels	Licznik_V=524
v_delay2	v_delay2	Licznik_V!=524

O przejściu do następnych stanów decyduje licznik *Licznik_V* narysowanych do tej pory linii poziomych, liczący modulo 525. W stanie *vh_sync_now* na wyjściu *vhsync* ustawiony jest stan niski.

5. Bibliografia

1. Projekt Krzysztofa Lisa „BMP2VGA” opisany na stronie:
<http://www.zpt.tele.pw.edu.pl/~ptomasze/ucyf/2005l/1/bmp2vga1.htm>
2. „Układy FPGA w przykładach” – Jacek Majewski, Piotr Zbysiński
3. „Układy programowalne. Pierwsze kroki” – Piotr Zbysiński, Jerzy Pasierbiński
4. „Projektowanie układów cyfrowych z wykorzystaniem języka VHDL” Mark Zwoliński
5. Wikipedia – historia i zasady gry snake