

## Praca w środowisku ISE c. d.

### a) Schemat logiczny

Tworzenie schematu logicznego projektu skutkuje wygenerowaniem pliku z rozszerzeniem .SCH. Do tworzenia schematów używa się edytora ECS.

Symbole, z których możemy korzystać podczas opracowywania projektów, są opisane w „Libraries Guide”. Elementy dzieli się na:

- **PRIMITIVES** – elementy podstawowe. Są to takie elementy, których nie można rozłożyć na czynniki pierwsze. Z takich elementów składają się elementy bardziej złożone, nazywane makrami.
- **SOFT MACROS** – makra ogólne, składające się z elementów podstawowych. Takim przykładem może być przerzutnik typu JK, który składa się z przerzutnika D oraz towarzyszących mu bramek logicznych.
- **RELATIONARY MACROS** – są to makra zawierające informacje o rozmieszczeniu elementów składowych. Jest to ważne w układach FPGA gdzie mamy do czynienia z dwuwymiarową matrycą i tam ważne jest odpowiednie rozmieszczenie względem siebie elementów. Makra z informacją o rozmieszczeniu nazywane są „Relationary Placed Macros”. Przykładem mogą być sumatory kaskadowe, składające się z łańcucha pojedynczych sumatorów. W układach CPLD informacje o rozmieszczeniu nie są potrzebne.

Na schemacie logicznym umieszcza się porty we/wy („I/O Markers”). Jeżeli dany schemat reprezentuje element szczytowy projektu, to porty muszą być przypisane do konkretnych wyprowadzeń układu. W przypadku hierarchicznej struktury projektu (gdzie dany schemat jest tylko podschematem) porty we/wy nie muszą być opisywane.

### b) Opis w języku HDL (Hardware Description Language)

Obecnie używa się dwóch języków opisu sprzętu: Verilog i VHDL. My będziemy zajmować się tylko językiem VHDL, który był tworzony przez kilka firm naraz. Miał służyć do opisu układów wielkiej skali integracji. Język VHDL ma więcej możliwości opisu i przez to jest trudniejszy do opanowania.

Opisu można dokonywać przy pomocy zwykłego edytora tekstowego z zachowaniem odpowiedniej składni. Pliki z kodem mają rozszerzenie .VHD.

Dostępne są gotowe szablony, które należy wykorzystywać podczas opisywania układów w języku VHDL z przeznaczeniem do syntezy. Ścieżka dostępu to VHDL → Synthesis Constructs → Coding Examples. Szablony podają sposoby opisywania rejestrów, transkoderów, multiplekserów, liczników... Korzystanie z szablonów jest wskazane dla powodzenia procesu syntezy.

Opisy sprzętu w postaci rysowania schematów i kodów języka VHDL nie są rozdzielne. Można je mieszać i robi się to. Dla danego modułu utworzonego w języku VHDL można utworzyć symbol, pozwalający na umieszczenie modułu na schemacie.

### c) Maszyny stanów (Finite State Machines FSM)

Przeznaczone do opisu maszyn stanów, które mają realizować algorytm sekwencyjny, opisany przy pomocy grafu stanów.

W pakiecie istnieje edytor „StateCAD” służący do rysowania grafu automatu. Pliki generowane przez edytor mają rozszerzenie .DIA.

Opis zawarty w pliku .DIA jest automatycznie konwertowany na opis maszyny w języku VHDL. Później można opracować symbol maszyny, który można umieścić na schemacie logicznym układu.

Istotną rzeczą jest opis wyprowadzeń układu. Zależą one od obudowy w jakiej zamknięto układ. Wykorzystuje się tutaj plik z ograniczeniami projektowymi (User Constraints File) z rozszerzeniem .UCF. Trzeba go dodawać do każdego projektu. Definiuje się w nim przypisanie portów we/wy do wyprowadzeń obudowy:

```
NET „NazwaPortu” LOC = „Pn”;
```

„NazwaPortu” opisuje port we/wy występujący na schemacie szczytowym projektu. Format numeru wyprowadzenia „Pn” oznacza zapis „P20” lub „P8”. Numer wyprowadzenia zależy od typu obudowy. Jeżeli będziemy pracować z obudową PLCC, to będziemy używać oznaczenia „P05”. Jeżeli trafilibyśmy na bardziej skomplikowaną obudowę (BGA – wyprowadzenia rozmieszczone pod układem w postaci matrycy 2-wymiarowej) to wy-

przewodzenia opisuje się podobnie jak na planszy szachowej: „A0”, „B15”. Numer wyprowadzenia musi być brany ze schematu układu zamkniętego w konkretnej obudowie. Na laboratorium pracujemy ze scalakiem zamkniętym w obudowie PLCC-44. Jeżeli zabraknie pliku .UCF, nie zostanie wygenerowane żadne ostrzeżenie. Porty zostaną przypisane automatycznie przez narzędzie na zasadzie „jak leci”.

#### d) Symulacja

Praca będzie się odbywała z symulatorem „ModelSIM”. Jest to oprogramowanie zewnętrzne, ale integruje się z pakietem ISE. Możemy przeprowadzać dwa rodzaje symulacji układów:

- **FUNKCJONALNA (Behavioral)** – jest prostą symulacją zero-jedynkową. Nie są uwzględniane żadne opóźnienia sygnałów. Jest prostsza do analizy i szybciej wykonywana przez program. Symulacja czasowa może iść w godziny przy bardzo złożonych projektach. Symulacja funkcjonalna pracuje dokładnie na stworzonym przez nas opisie zanim został przeprowadzony proces implementacji. Umożliwia nam ona śledzenie sygnałów wewnętrznych projektu. Definiujemy pobudzenia dla portów wejściowych i zostają obliczone odpowiedzi dla portów wyjściowych.
- **CZASOWA (Post-Fit)** – Tu również można śledzić sygnały wewnętrzne, lecz są one już zsyntezowane przez narzędzie i mogą być one inne, niż przez nas ustalone, bo sygnały mogły zostać zoptymalizowane podczas procesu sklejania logiki.

Każda symulacja musi się rozpocząć od zdefiniowania pobudzeń czasowych na portach wejściowych. Definicje przechowywane są w pliku „Testbench” który może być tworzony edytorem graficznym (kształtowanie fal prostokątnych poprzez klikanie myszą) generującym pliki .TBW. Pobudzenia mogą być przechowywane w pliku VHDL, gdzie musimy definiować procesy podające pobudzenia na wejścia naszego projektu.

#### e) Konfiguracja układu

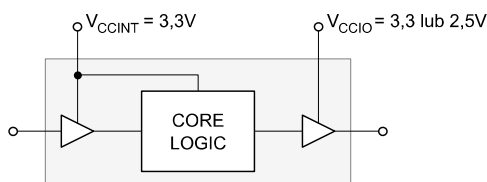
Jest to ostatni krok polegający na przesłaniu zaimplementowanego projektu do układu. Plik z danymi konfiguracyjnymi dla układów CPLD ma rozszerzenie .JED. Pliki konfiguracyjne dla scalaków FPGA mają rozszerzenie .BIT. Do przesyłania używa się programu „Impact” oraz kabla programującego dołączonego do portu LPT w komputerze. Dane konfiguracyjne są po stronie układu dostarczane przez interfejs JTAG.

*[Wracamy do omawiania architektury układów 9500]*

#### Bloki WE/WY

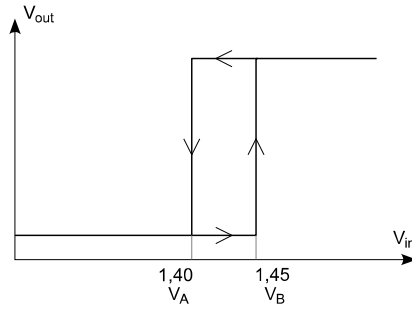
Każdy z bloków jest skojarzony z jedną makrokomórką by mógł wyprowadzać sygnał logiczny obliczony w danej makrokomórce. Sygnał wejściowy najpierw trafia do bloku we/wy, później do globalnej matrycy połączeń by w końcu trafić do makrokomórki.

Wspomnieć jeszcze należy o sposobie zasilania układów 9500XL, które jest podwójne:

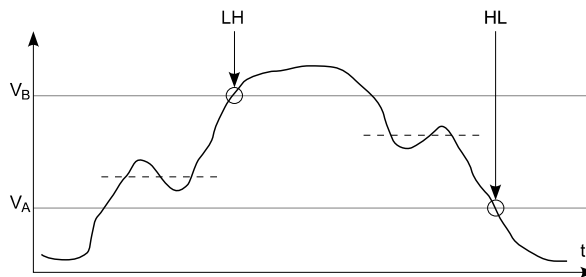


Logika wewnętrzna oraz bufor wejściowe są zasilane napięciem  $V_{CCINT}$  oraz bufor wyjściowe zasilane są napięciem  $V_{CCIO}$ .  $V_{CCINT}$  musi być zawsze równe 3,3V. Jeżeli chodzi o  $V_{CCIO}$ , to możemy podawać napięcie 3,3V lub 2,5V. Niższe napięcie przydaje się wówczas, gdy na wyjście układu 9500 podłączy się niskonapięciowe układy zasilane napięciem 2,5V.

Jeżeli dany pin pracuje jako wejście, to automatycznie zostaje wyłączona ścieżka wyjściowa w danym bloku. Sygnał przechodzi przez bufor wejściowy i trafia do matrycy Fastconnect. Napięcie 3,3V zasilające bufor wejściowy wynika ze stałego progu przełączania takiego bufora. Konstrukcja bufora jest tak pomyślana, by występowała tam mała pętla histerezy:



Napięcia określające zakres histerezy są ustalone na poziomie 1,40 oraz 1,45V. Gdy napięcie na pinie rośnie, to by na wyjściu pojawiła się „jedyńka” napięcie musi przekroczyć próg 1,45V. Gdy napięcie będzie spadało, „zero” logiczne pojawi się dopiero po przekroczeniu progu 1,40V. Takie formowanie sygnałów logicznych jest potrzebne. Ma to uodpornić piny wejściowe na fluktuacje napięcia wejściowego. W technologii TTL stosowało się przerzutniki Schmitta z szeroką pętlą histerezy.



Pętla przydaje się gdy sygnał wejściowy zmienia się w czasie. Szeroka pętla histerezy sprawi że odfiltrowane są wahania sygnału wejściowego; zapobiega ona wielokrotnym przełączaniom między zerem a jedynką w przypadku zboczy narastających wolno i z zakłóceniami. Gdybyśmy używali bramki z jednym progiem dyskryminacji (poziome linie przerywane), to każde przejście sygnału wejściowego przez próg powodowałoby przełączanie się bramki.

Ważną własnością jest tolerowanie napięć wejściowych do wartości 5V (zakres TTL). Sam układ pracuje z napięciem 3,3V. Podanie na wejście układu CMOS napięcia wyższego, niż napięcie zasilające powodowało spalanie układu. Poza tym konieczne było zastosowanie specjalnych procedur podłączania napięć zasilających. Układy 9500 są odporne na zjawisko „Power Sequencing” (kolejność podawania napięć zasilających). Układy CMOS bez zabezpieczeń wymagają ostrożnego podawania napięcia zasilającego.

Po stronie wejściowej istnieje zgodność z różnymi rodzinami układów. Najwyższe napięcia są generowane przez układy TTL i 5V układu CMOS. Można dołączać układy CMOS zasilane napięciami 3,3V oraz 2,5V. Z tego powodu wzięto się napięcie progowe na poziomie 1,45V.

### Bufor wyjściowy

Zasilanie bufora napięciem na poziomie 2,5V podyktowane jest sterowaniem przez bufor układów CMOS zasilanych napięciem 2,5V, które nie tolerują wysokich napięć na swoich wejściach. Układy 9500 stosuje się jako interfejs logiki niskonapięciowej. Jeżeli bufor będzie zasilany napięciem o wartości 3,3V, to mamy zapewnioną jego zgodność z układami TTL (i CMOS zasilanych napięciem 5V). Jeżeli  $V_{CCIO} = 2,5V$ , to pojawią się kłopoty ze współpracą z układami TTL i CMOS'ami zasilanymi 5V. Oczywiście zgodność z układami 2,5V jest zachowana.

Bufor wyjściowy oferuje opcję obniżenia szybkości narastania zboczy sygnału (Slew rate). Regulacja szybkości jest tu dwustopniowa. Standardowo wyjście działa z szybkim narastaniem (opadaniem) zboczy. Zmiany prędkości narastania zboczy dokonać można przez użycie atrybutu SLEW w opisie portu wyjściowego. W pliku UCF będzie to wyglądało tak:

```
NET „Cośtam” LOC = „Coś” | FAST{SLOW};
```

Obniżanie szybkości przełączania stosuje się celem odfiltrowania impulsów prądowych powstających podczas jednoczesnego przełączania wielu wyjść naraz. Problem ten daje się we znaki w systemach magistralowych w czasie podawania adresu. Każde zbocze (przełączanie się bufora) generuje impuls prądowy. Dla pojedynczego

bufora przy szybkim zboczu może wystąpić impuls prądowy o wartości nawet 10mA. Jeżeli mamy kilkaset buforów i kilkadziesiąt będzie się przełączało naraz, to impuls prądowy jest już rzędu amperów. Poza tym musimy mieć zasilacze, które dadzą sobie radę z dostarczaniem odpowiedniej ilości energii oraz dobrze poprowadzone wyprowadzenia GND, ponieważ impuls prądowy może wywołać skok wartości potencjału masy, co może spowodować zakłócenia w pracy układu. Lepsze jest zatem wydłużenie czasu narastania zboczy by zmniejszyć impulsy prądowe. W układach FPGA regulacja prędkości narastania zbocza ma większą możliwość konfiguracji.

W bloku wyjściowym mamy różne źródła sygnału Output Enable. Może być on pobierany jako term z makrokomórki (PTOE), może to być również jeden z sygnałów GTS lub nawet wartość stała (0 lub 1).

Istnieje tu coś takiego jak User Programmable Ground, układ sprawiający że nieużywany pin może pełnić rolę dodatkowego wyprowadzenia masy GND. Przydaje się to wówczas gdy mamy wiele przełączających się wyjść i dodatkowa masa może tłumić w pewnym stopniu impulsy.

Bus Hold („Pin Keeper” w środowisku ISE). Jest to coś na kształt asynchronicznego przerzutnika, który ma za zadanie podtrzymywanie stanu sygnału na wyjściu.

Istnieje jeszcze układ Pull-Up (oparty na rezystorze  $R_{BH} \approx 50k\Omega$ ) który sprawia, że pin jest podłączony przez rezystor do jedynki logicznej. Ma to zapobiec samowolnemu przełączaniu się układu (nie sprzęga się z otaczającą logiką, nie łapie zakłóceń).