



Laboratorium 4

Verilog – Liczby BCD

Instytut Telekomunikacji Multimedialnej

Skład Grupy: Bartosz Gabruk 157546

Data: 06.04.2025r.

4. Czy układ działa poprawnie? Sprawdzić z innymi wartościami liczb poddawanych konwersji: 16, 127..... (1p)

Tak, ten układ prawidłowo realizuje konwersje do BCD dla podanych liczb.

5. Dlaczego moduł w takiej postaci nie jest dobrą implementacją sprzętową? (2p)

To rozwiązanie posiada brak synchronizacji z zegarem (nie działa krok po kroku tylko natychmiast przepisuje wynik). Operacje dzielenia i reszty z dzielenia są słabym rozwiązaniem ponieważ są kosztowne w zasoby logiczne sprzętu oraz są powolne. Wyniki O_DAT i O_STB – nie są przechowywane w rejestrach, nie kontrolujemy ich przebiegów. Ten kod także nie skaluje się dobrze dla większych danych (więcej niż 32 bity)

9. Czy układ działa poprawnie? Sprawdzić z innymi wartościami liczb poddawanych konwersji: 16, 127..... (1p)

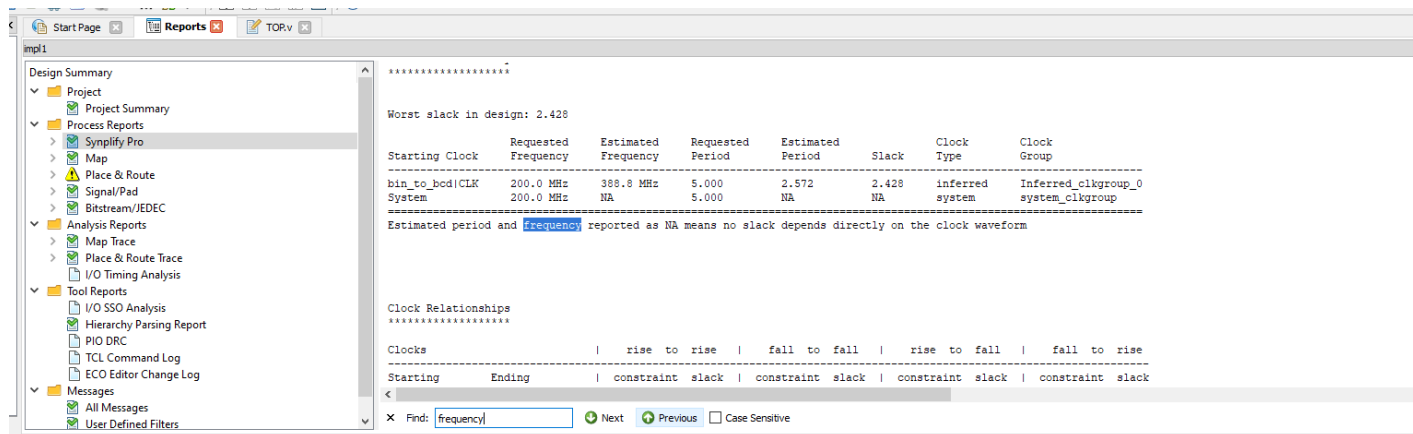
Układ działa poprawnie tylko dla jednocyfrowych liczb. Dla podanych wyżej (16, 127) nie konwertuje ich poprawnie do BCD.

10. Przyjrzeć się przebiegom czasowych linii „bcd0””bcd9” i znaleźć na nich przyczynę nieprawidłowego działania modułu. Co należy zmienić? (3p)

Przyczyną błędu jest przepełnienie (overflow) przesunięte o jeden takt. W module bcd_shl_1 sygnał overflow jest aktualizowany wewnątrz bloku always razem z rejestrowanym DAT. W efekcie następuje przesunięcie o jeden takt wartości przepełnienia.

Należy zmienić sposób przypisania overflow – przenieść go poza blok always i wyprowadzić jako sygnał kombinacyjny (assign),

20. Jaka jest maksymalna częstotliwość pracy układów? (1p)



388.8 MHz (estimated frequency)

Który okazał się mniejszy i dlaczego?

(1p)

Mniejszy pod względem wykorzystania zasobów logicznych okazał się układ, który wykorzystuje moduł `bin_to_bcd`. Przez to, że nie używa kosztownych operacji dzielenia i modulo, lecz jedynie proste przesunięcia i overflow jest obsługiwany przy użyciu `assign`, zamiast być rejestrowanym wewnątrz bloku `always`, co umożliwiła syntezę przy niewielkim zużyciu rejestrów. Ta wersja jest mniejsza pod względem zasobów (użyto tylko 147 rejestrów (147 of 4320 (3%)) i 107 LUTów)

```
Finished technology mapping (Real Time elapsed 0h:00m:01s; CPU Time elapsed 0h:00m:01s;
Memory used current: 174MB peak: 174MB)

Pass          CPU time          Worst Slack          Luts / Registers
-----
1             0h:00m:01s       1.95ns               107 / 147
```

Czy implantacja z punktu 2 się syntezyje? Jeśli nie to dlaczego?

(1p)

„`Bin_to_bcd_simple`” nie syntezyje poprawnie – problemem są w niej operacje dzielenia i modulo, które nie są wspierane przez narzędzia syntezy.