**Zajęcie Nr1 – PODSTAWY JAK ZBUDOWAĆ ARDRUINO**

Dokonując każdej operacji na płytkach i podzespołach musimy dotknąć masy (w tym przypadku dowolnego elementu metalowego)

Czym jest to środowisko? – Środowisko w którym między innymi AMD projektuje procesory

Lokacje dla plików na zajęciach:  
/HDL/PROJEKTY/G5\_N - przygotowane przez nas pliki

/HDL/DSW/PROJEKTY/ - gotowe wcześniej pliki

Język skryptowy TCL jest na górze projektów jest specyficzny, z definicji systemy operacyjne nie lubią znaków białych, znaków specjalnych – należy pamiętać o tym przy tworzeniu projektów

Constrains – ograniczenia, jakie napięcia wyjściowe i wejściowe mając zostać podawane do układu. HLD developer z góry definiuje jakie wejścia i wyjścia funkcjonują i w jaki sposób mają funkcjonować.

SpeedGrade – szybkość pracy w domenach (to nie jest minus, to jest myślnik)

Dodane przez nas pliki z /HDL/DSW/PROJEKTY/ mają oznaczenie trzech kwadracików – jest w trybie „top module” i nie powinien się on odpalać. Trzeba pamiętać że jeden moduł jeden plik, ale projekty mogą być wielomodułowe, TOP MODULE to plik główny który odpala resztę procesów.

Nie zawsze środowisko zrobi nam TOP MODULE automatycznie, nie należy się na to zdawać i zawsze trzeba oscenić jaki plik zadziała jako TOP MODULE.

(dla chętnych, instrukcja piąta to instrukcja jak zbudować własne ardruino)

100ns – 3 cykle

„czy jest różnica między licznikiem binarnym a dzielnikiem częstotliwości”

Jedna fala 50hz daje nam na bicie 0 25hz, na kolejnym bicie daje 12,5hz

NIE, nie ma różnicy między licznikiem binarnym a dzielnikiem częstotliwości, jest on naturlanym dzielnikiem częstotliwości

(największy zarobek jest przy IP Core)

Żeby zrobić implementację:

- kliknąć w „IP Inedgrator”, a następnie wybrać „Create block design”

- klikamy tools i wybieramy „create and package new ip”

- ZAWSZE „create your current project”

- wybieramy ścieżkę /HDL/PROJEKTY/G5\_N/PROJEKT\_1/PROJEKT\_1.SRC/SOURCES\_1/IMPORTS/IP/BLINK\_IP i pozwalamy by stworzył ścieżki

- klikamy finish

- wtedy zgłasza się druga instancja vivado, poprzedniej nie zamykamy,

- wybieramy Reviev and package w opcji „package steps”

- wybieramy „package ip”

- wracamy do starej instancji vivado

- aby sprawdzić czy modul działa, klikamy w plusa nad pusta kartka i wpisujemy nazwe ip, nie zatwierdzając

- ZAWSZE BUDOWANIE ZACZYNAMY OD DOMENY ZEGAROWEJ „clock wizard”

- po ustawieniu zegara „run connection automation”

- nasz system w constrain będzie miał automatycznie podany port wyjściowy zegara

- dodać ip core projektu i połączyć z tak stworzonym zegarem

- ctrl+s

- ctrl+t z wybranym modułem tworzy port wyjściowy, taki port wyjściowy jest NIEZBĘDNA

- kliknąć prawym na blok i wybrać „generate output products” dla

- kliknąć prawym przyciskiem na cały design i wybrać „create HDL wrapper”

JAK WYGLĄDA ZALICZENIE?

Zbudować samodzielnie od zera projekt z licznikiem, otrzymamy iformacje z kąd wziąć constrains

- musimy zbudować tyle ipcore ile jest niezbęde,

- nazwać block design tak jak jest w projecie – constrains są nszykowane

Zaliczenie dopiero gdy pokażemy jak zmodyfikować taki kod żeby działał

TO JEST NA 3

-add sources

-constrain

- project.sources/constrains/new