

**Przetwornik Sigma-delta na układzie FPGA**

Brak tego projektu w wykazie projektów na UPEL-u.

Etapy projektowe które wymaga prowadzący (odpowiadają poszczególnym sprawozdaniom):

1. **Zgłoszenie tematu, podział na grupy projektowe, opis algorytmu**

Zrobione

1. **Model behawioralny algorytmu**

Poziom behawioralny lub algorytmiczny (ang.behavioral level). Jest to najwyższy poziom abstrakcji dostępny w Verilogu. Opisywanie układu na tym poziomienie jest zależne od szczegółów technologicznych i przypomina programowanie wjęzyku wysokiego poziomu.

Jak wygląda przykładowy model behawioralny? – strona 47 <http://staff.uz.zgora.pl/rwisniew/instrukcje/inne/verilog/verilog_kurs.pdf>

Prowadzący wymaga jedynie opisu samego działania algorytmu nie koniecznie w behawioralnym HDL, może to też być C, Python, Mathlab!

1. **Struktura procesora dedykowanego**

Cyt.: „Proszę o przygotowanie ogólnej struktury sprzętowej proponowanego procesora. Schemat blokowy powinien wykorzystywać takie elementy składowe jak porty we/wy, rejestry, multipleksery, pamięci, bloki kombinacyjne. Proszę, aby ze struktury w sposób oczywisty wynikał sposób integracji procesora z magistralią AXI i komuinikacja z CPU.

Schemat proszę załączyć jako jeden plik graficzny (najlepiej JPG), a ewentualny komentarz dodać w txt. Jeżeli arkuszy schematu będzie więcej proszę użyć PDF.”

Pyt:

Jak ułożyć dane elementy nie mając gotowego kodu, przecież to właśnie z implementacji będzie wynikać rozmieszczenie tychże podzespołów?

CPU jest już zaimplementowany?

1. **Syntezowalny model HDL procesora**

Cyt: „Proszę załączyć w txt:

1. syntezowalny kod procesora w HDL,

2. kod testbench,

3. wyjście na konsolę z testbench,

4. wyciąg z raportu syntezy układu - liczbę przerzutników i elementów LUT,

5. dodatkowy opis i komentarz jeżeli konieczne.

Proszę tak przygotować testbench, aby bez koniecznego opisu można było na podstawie wyjścia na konsolę ocenić poprawną pracę procesora.”

Pyt.:

Syntezowalny kod? Czy to jest po prostu kod w HDL?

1. **Budowa AXI IPCore procesora dedykowanego i integracja z procesorem ogólnego przeznaczenia**

Cyt: „Proszę przygotować syntezowalny kod procesora dedykowanego współpracujący z magistralą AXI.

W sprawozdaniu proszę załączyć jako txt:

1. Krótki opis sposobu komunikacji CPU (Zyng ARM, Microblaze) z procesorem. Proszę napisać, czy komunikacja przebiegać będzie za pośrednictwem rejestrów IO, FIFO, a może DMA. Opisać przestrzeń adresową IO procesora na AXI i podać funkcję poszczególnych adresów w pamięci.

2. Kod HDL integrujący AXI IP core z kodem procesora dedykowanego. Szablon tego kodu jest typowo generowany przez Vivado. Państwa zadanie polega na podłączeniu do tego szablonu swojego procesora. W razie wątpliwości proszę zerknąć do ćwiczenia 2 z laboratorium, gdzie integrowany jest procesor CORDIC.

3. Raport o zużywanych zasobach FPGA po syntezie. Proszę połączyć AXI IP core z procesorem CPU i dokonać jego syntezy pod platformę Zedboard (jak w ćwiczeniu 2 i 3 realizowane było dla Cordic).

W razie wątpliwości służymy konsultacjami online lub na email.”

Plan działania:

1. Poznać sposób działania przetwarzania sigma-delta
2. Przygotowanie algorytmu w języku C, Python lub Matlab algorytmu sigma-delta
   1. Kod powinien być bardzo dobrze opisany – komentarze
   2. Kod powinien mieć adekwatne nazwy zmiennych do zastosowania
3. Budowa graficzna algorytmu i przedstawienie jej w blokach
   1. Jakie elementy są niezbędne?
   2. Dane wejściowe – skąd? Jakie?
   3. Dane wyjściowe – porty propagacja
   4. Integracja z AXI
   5. Komunikacja z CPU
4. Przenieść dwa poprzednie punkty na Verilog
   1. Napisać kod algorytmu
   2. Napisać testbench
   3. Zrobić krótki raport
5. Będzie czas martwić się tym punktem

Trochę teorii odnośnie samych przetworników sigma-delta:

<https://elektronikab2b.pl/technika/12097-sigma-delta.-przetworniki-analogowo-cyfrowe-czasu-ciaglego>

Podstawy Veriloga - <http://fiona.dmcs.pl/~cmaj/Verilog/Kurs%20verilog.pdf>

<http://staff.uz.zgora.pl/rwisniew/instrukcje/inne/verilog/verilog_kurs.pdf> - politechnika zielonogórska

wyjaśnienie podstawowych elementów języka - <https://neo.dmcs.pl/jhdl/private/verilog.pdf>