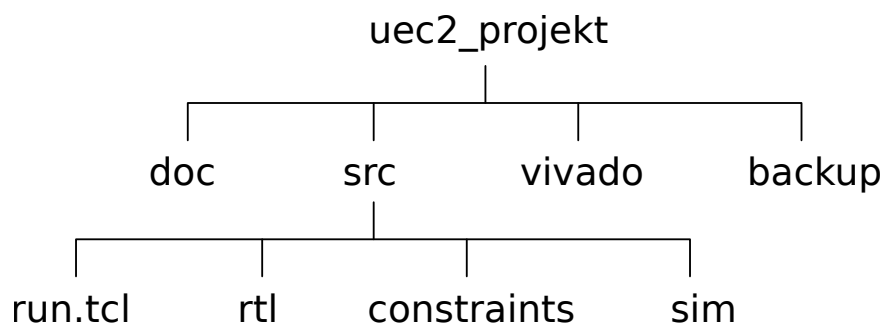


1. Wymagania ogólne

1. Projekty wykonujemy w zespołach 2 osobowych (indywidualne wykonanie projektu jest również akceptowane).
2. Projektowane urządzenie/gra musi:
 - 2.1. posiadać interfejs użytkownika (mysz, klawiaturę, lub inne urządzenie zewnętrzne)
 - 2.2. wyświetlać wyniki na ekranie VGA o rozdzielczości 1024 x 768 lub zbliżonej,
 - 2.3. być zaimplementowane na 2 płytkach uruchomieniowych Basys 3 z komunikacją między nimi; czyli multiplayer, jeżeli to jest gra; każdy użytkownik ma mieć swój interfejs podłączony do własnej płytki (ekran, mysz, etc.),
 - 2.4. projekty nie spełniające tych wymagań muszą dostać wyraźną zgodę prowadzącego.
3. Program ma być napisany w języku Verilog v2001, zgodnie z podanymi regułami projektowania. Inne style kodowania mogą być stosowane tylko w oparciu o udokumentowane źródła (np. UG XST firmy Xilinx).
4. Do pisania kodu modułów, poza uzasadnionymi przypadkami używamy dostarczonych wzorców projektowych.
5. Wszystkie rejestry w projekcie powinny być kasowane **resetem synchronicznym**. Reset asynchroniczny jest dopuszczony tylko w kontrolerze resetu.
6. Struktura katalogów projektu powinna być następująca:



Podkatalogi:

- doc – dokumentacja i raporty,
- src – wszystkie pliki potrzebne do utworzenia projektu Vivado
- run.tcl – skrypt pozwalający na utworzenie projektu, przeprowadzenie symulacji, generację bitstreamu oraz jego wgranie na FPGA
- rtl – pliki potrzebne do syntezy projektu
- constraints – pliki konfiguracyjne .xdc
- sim – pliki potrzebne do symulacji

- vivado – katalog projektu vivado (skrypt run.tcl powinien uruchamiać program Vivado w tym folderze, a wszystkie generowane wskutek tego foldery i pliki, w tym logi, powinny się tu znaleźć)
- backup¹ – katalog na archiwizowanie projektu na różnych etapach realizacji;

7. inne podkatalogi mogą być tworzone w miarę potrzeb

8. należy użyć systemu kontroli wersji **git** i odpowiedniego serwisu np. <https://github.com> lub <https://bitbucket.org>. Użycie gita (rzeczywiste, nie pozorne), będzie punktowane.

Książka do GITa: <https://git-scm.com/book/pl/v2>

9. W repozytorium należy przechowywać wyłącznie zawartość folderu src. Zawartość repozytorium powinna pozwolić na skompilowanie projektu.

10. Można korzystać z modułów zewnętrznych, np. ściągniętych z sieci, pod warunkiem że są poprawnie zakodowane. Należy wyraźnie wskazać źródła ich pochodzenia (link w nagłówku pliku).

11. Do oceny należy spełnić poniższe wymagania:

11.1. Załadować na UPEL 3 (trzy) pliki:

1. raport – plik **PDF** (inne formaty nie będą akceptowane!);
2. projekt – plik **ZIP** (inne formaty nie będą akceptowane) - katalog src, ostatnie archiwum z katalogu backup, oraz **bitstream do zaprogramowania płytki BASYS3**;
3. wypełnioną listę kontrolną (checklist) wg załącznika – plik **PDF**;
projekty bez tej listy nie będą sprawdzane! (ocena = 0%)

11.2. Zaprezentować działanie wykonanego projektu na jeden z poniższych sposobów

1. podczas ostatnich zajęć projektowych zaprezentować film lub działania projektu na forum roku, opowiedzieć skąd wziął się pomysł, co było trudne / łatwe / ciekawe / nudne...
2. przygotować film demonstrujący działanie programu; film powinien być krótki (1-2 minuty), i zawierać odniesienie do kierunku MTM oraz strony www.mtm.agh.edu.pl; link do filmu umieszczonego w chmurze należy umieścić w raporcie,

2. Realizacja

1. Uzgodnić temat projektu z prowadzącym. Po zatwierdzeniu temat staje się obowiązujący. W przypadku powtarzających się tematów pierwszy przysłany wygrywa. Tematy mogą być modyfikowane przez prowadzącego.

Osoby/zespoły, które nie zaproponują tematu, otrzymają tematy przydzielone przez prowadzącego

¹ Tworzenie backupów nie jest konieczne w przypadku korzystania z gita

2. Proszę planować zakończenie projektu do 1 czerwca 2020 (wtedy jest szansa, że zostanie on oddany przed wakacjami)

3. Ocena

1. Oceniane będą tylko projekty z wypełnioną listą kontrolną.

3.1. Elementy oceny

1. Poprawność raportu finalnego i bazy projektu: 20%

- 1.1. zgodność z wymaganiami.

2. Poprawność implementacji: 30%

- 2.1. jednolity i poprawny styl kodowania; każdy moduł ma mieć oznaczonego w nagłówku autora,
- 2.2. hierarchia projektu zgodna z dokumentacją,
- 2.3. brak ostrzeżeń krytycznych i błędów,
- 2.4. poprawne parametry czasowe (setup i hold),
- 2.5. sensowne wykorzystanie zasobów, np. używanie globalnych buforów i modułów DSP tylko tam, gdzie są one rzeczywiście potrzebne.

3. Poprawne wykorzystanie systemu kontroli wersji GIT: 20%

- 3.1. przez wszystkich członków zespołu ,
- 3.2. w ciągu całego czasu trwania projektu,
- 3.3. wykorzystanie rozgałęzień (branch),
- 3.4. w repozytorium mają być tylko pliki źródłowe (nie generowane).

4. Poprawność działania programu: 10%

- 4.1. działanie programu zgodnie ze specyfikacją.

5. Subiektywna ocena prowadzącego: 20%

- 5.1. m. in. stopień trudności, atrakcyjność tematu, wrażenie itp.

6. Prezentacja / Film prezentujący projekt: 10%

- 6.1. zgodność z wymaganiami, jakość wykonania, czy prezentuje wszystkie funkcje.

UWAGA: filmy będą udostępnione wszystkim studentom na roku po wystawieniu ocen.

Autorów najlepszych filmów będziemy prosić o zgodę na ich wykorzystanie do reklamowania kierunku MTM.

7. **Bonus: +10% extra** za oddanie projektu w pierwszym terminie. Bonus otrzymuje się, jeżeli ocena projektu bez bonusu wynosi co najmniej 80%.

4. Terminy

1. Ostateczny termin ustalenia tematu: **16 maja 2022**
2. Pierwszy termin oddania projektu: **20 czerwca 2022.**
3. Ostateczny termin oddania projektu: **15 września 2022.**

UWAGA: Po ostatecznym terminie nie będą przyjmowane projekty **z żadnych powodów**, w tym:

- „zapomniałem załadować listy kontrolnej ...” ,
- „spakowałem projekt rarem, bo lepiej kompresuje”, (BTW jaką rar ma licencję?),
- „właśnie przez przypadek skasowałem sobie projekt”,
- „internet nie działał i nie mogłem załadować na UPEL”,
- „nie zdążyłem do końca zrobić ...”,
- „zapomniałem załadować kilku plików ...”.