

Systemy komputerowe: architektura i oprogramowanie (SYKOM25L) - Projekt

Politechnika Warszawska, Instytut Telekomunikacji

Prowadzący: Aleksander Pruszkowski

Osoba dla której przygotowano ten dokument Krolak-Kurkus Wojciech

Zadanie projektowe: Należy realizować korzystając z tzw. wirtualnych komputerów rezerwowanych za pomocą serwera WWW <https://resrepo.tele.pw.edu.pl>. Rezerwacje dla projektu są dokonywane w slotach maksymalnie 2h. Maszyny można rezerwować wielokrotnie lecz liczba rezerwacji dla poszczególnych etapów jest ograniczona. Projekt realizowany jest indywidualnie (grupa jedno osobowa)!

Zgodnie z opisem wprowadzającym SYKOM_proj.pdf proszę utworzyć:

- Moduł Verilog w pliku: gpioemu.v, który ma realizować operację wyznaczania sumy kontrolnej opisywanej w literaturze jako: CRC32ISCSI. Suma ta ma być obliczana z treści zapisanej w bloku pamięci przechowywanym w wnętrzu modułu verilog, blok ten ma mieć wielkość 250 lokacji 8 bitowych.

Dane użytkownika mają być wprowadzone pojedynczo do tego bloku (po 8 bitów) poprzez rejestr wejściowy IN ulokowany pod adresem relatywnym: 0x640 w przestrzeni pamięciowej procesora. Użytkownik gdy uzna, iż zappełnił blok danymi i chciałby aby moduł wykonał obliczenie sumy kontrolnej przekazuje poprzez rejestr CTRL umiejscowiony pod adresem relatywnym 0x658, wartość będącą odpowiednio zaprojektowanym poleceniem, innymi poleceniami mogłyby być np.: wyczyść cały blok, ustaw pozycję zapisywania danych użytkownika w wnętrzu bloku na pozycję 0, itp. W raporcie przedstaw analizę przyjętej przez Ciebie listy wspieranych poleceń i sposób ich bitowego kodowania.

Stan modułu np.: zappełnienie bloku, zakończenie liczenia sumy kontrolnej i innych podobnych sytuacji, ma być udostępniany procesorowi poprzez rejestr STATE ulokowany pod adresem relatywnym 0x648 w jego przestrzeni pamięciowej.

Za każdym razem gdy moduł poprawnie obliczy sumę kontrolną, jej wynik ma być dostępny dla procesora poprzez ulokowany pod adresem relatywnym 0x650 rejestr RESULT.

Pamiętaj aby moduł Verilog był zapisany optymalnie (m.in.: usuwaj z jego kodu to co nie potrzebne, co nie jest zapisane w tej specyfikacji lub jest w tym kodzie poprostu nadmiarowe). W kodzie tego modułu nie stosuj żadnych zewnętrznych bibliotek, nie stosuj także żadnych postaci pętli a kod Verilog niech wykorzystuje wyłącznie tzw. automaty stanów. Stopień spełnienia tych wymagań będzie oceniany. Nie zapomnij przetestować pliku Verilog odpowiednim (zgodnie z zasadami przekazanymi na laboratorium nr. 2 plikami testującymi które umieść także na GIT a ich działanie przedstaw w raporcie.

- Dla systemu Linux utwórz odpowiedni moduł jądra który udostępni użytkownikowi dostęp do rejestrów modułu Verilog za pomocą systemu plików SYSFS tak aby rejestry te dostępne były w tym systemie plików jako: IN: dskrwo, CTRL: dtkrwo, STATE: dckrwo, RESULT: drkrwo. Przyjmij, że dane przekazywane poprzez te pliki są w tekstowym formacie OCT. Pamiętaj o należytej obsłudze błędnych danych przekazywanych podczas współpracy z użytkownikiem czy błędnych sytuacji pojawiających się podczas pracy tak utworzonego systemu.
- Aby łatwiej testować i pokazać poprawność działania zbudowanego systemu - w szczególności ukazać zależności czasowe - proszę utworzyć w języku C aplikację przygotowaną do pracy w zbudowanym systemie Linux. Aplikacja ta w postaci binarnej ma być wbudowana w docelowy główny system plików (czyli w plik: rootfs.ext2) i ma ona po uruchomieniu przez użytkownika automatycznie demonstrować działanie systemu w różnych scenariuszach testowych. Pamiętaj, że podejście w testowaniu z wykorzystaniem systemowych poleceń cat i echo choć może zastąpić konieczność tworzenia powyższej aplikacji w C jest jednak oceniane niżej.

Zadbaj aby na GIT znalazły się wszystkie utworzone przez Ciebie artefakty, jednakże nie przenoś do tego repozytorium zbędnych elementów, m.in. produktów kompilacji, czyli produktów tworzonych przez udostępnione narzędzia. Spełnienie także tych wymagań będzie oceniane.

Zawartość raportu: Raport powinien na zamieszczonych w nim ilustracjach lub wycinkach logów wyświetlanych na ekranie QEMU ukazywać działanie systemu w różnych ważnych(!) chwilach działania systemu - dobór tych informacji będzie oceniany, jest on dowodem, że Autor jest pewien poprawności działania utworzonego przez siebie systemu i potrafi wykazać tą poprawność.

Raport (sprawozdanie z realizacji projektu) proszę umieścić na przydzielonym repozytorium GIT w formacie PDF. Żadne inne formaty dokumentów elektronicznych np.: DOC, DOCX, MD, ... nie będą przyjmowane. Fianlnie raport oraz wszelkie pliki źródłowe będące wynikiem prac nad projektem proszę umieścić w katalogu projekt repozytorium GIT - z tego (i tylko z tego!) miejsca prowadzący będzie pobierał pliki do późniejszego ocenienia i wystawienia oceny.

W raporcie oraz w kodach źródłowych wszelkie zapożyczenia wyraźnie zaznacz informując czytelnika skąd zostały one pobrane. Podczas zaliczania zadania projektowego pamiętaj, że obowiązuje Ciebie dokładna znajomość Twojego raportu i wytworzonych przez Ciebie kodów źródłowych.