

Systemy komputerowe: architektura i oprogramowanie (SYKOM) Projekt

Jakub Sulikowski

Numer albumu 318839

Prowadzący

mgr inż. Aleksander Pruszkowski

Warszawa, 15.05.2023

Spis treści

1. Wstęp	3
2. Przestrzeń adresowa	3
3. Testowanie modułu verilog	3
3.1 Test 1: Poprawność wyniku mnożenia oraz zliczania liczby jedynek	4
3.2 Test 2: Wynik nie mieści się na 32 bitach	4
4. Testowanie systemu	5
4.1 Uruchomienie systemu operacyjnego oraz dodanie modułu do jądra Linux	6
4.2 Testy w konsoli	7
4.3 Testy wykonane aplikacją użytkownika	9

1. Wstęp

Zadanie projektowe polegało na zbudowaniu układu z peryferiami wytworzonymi przez siebie i emulowanymi przez odpowiednio zmodyfikowany program QEMU. Następnie należało przetestować działanie tego systemu, z wykorzystaniem tworzonej dla tego środowiska dystrybucji systemu Linux oraz odpowiednich sterowników systemowych. Układ ten realizuje operację mnożenia przez dodawanie dwóch liczb 24-bitowych bez znaku.

2. Przestrzeń adresowa

Przestrzeń adresowa Gpio oraz rejestrów udostępnianych przez GpioEmu ma się następująco:

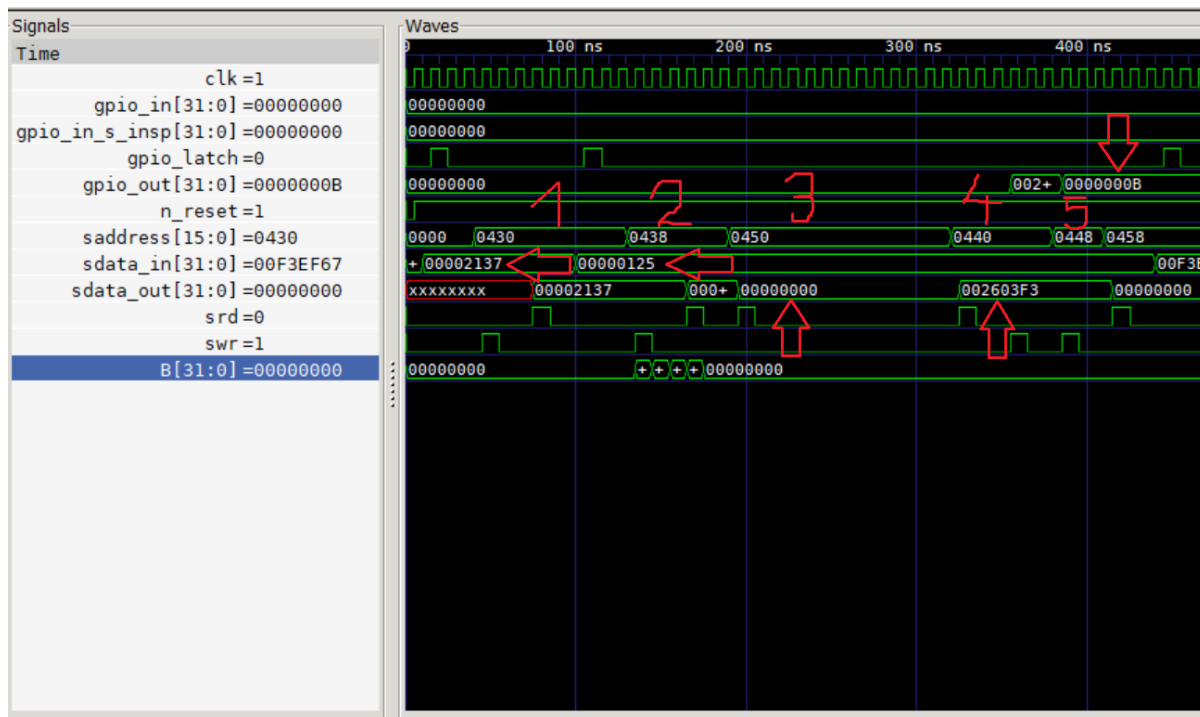
1. SYKT_GPIO_ADDR_SPACE - 0x80000790
2. SYKT_GPIO_ADDR_SPACE + 0x430 – Rejestr GpioEmu A1
3. SYKT_GPIO_ADDR_SPACE + 0x438 – Rejestr GpioEmu A2
4. SYKT_GPIO_ADDR_SPACE + 0x440 – Rejestr GpioEmu W
5. SYKT_GPIO_ADDR_SPACE + 0x448 – Rejestr GpioEmu L
6. SYKT_GPIO_ADDR_SPACE + 0x450 – Rejestr GpioEmu B

Zgodnie z zaleceniem zmieniono lokalizację adresu A1 z 0x42F na 0x430, tak by był podzielny przez 4.

3. Testowanie modułu verilog

Poprawne działanie modułu verilog zweryfikowano przy pomocy narzędzia GTKWAVE. Sprawdzone czy moduł poprawnie wykonuje operację mnożenia, zliczania jedynek. Dodatkowo sprawdzono czy moduł reaguje na nie mieszczące się wyniku na 32bitach.

3.1 Test 1: Poprawność wyniku mnożenia oraz zliczania liczby jedynek



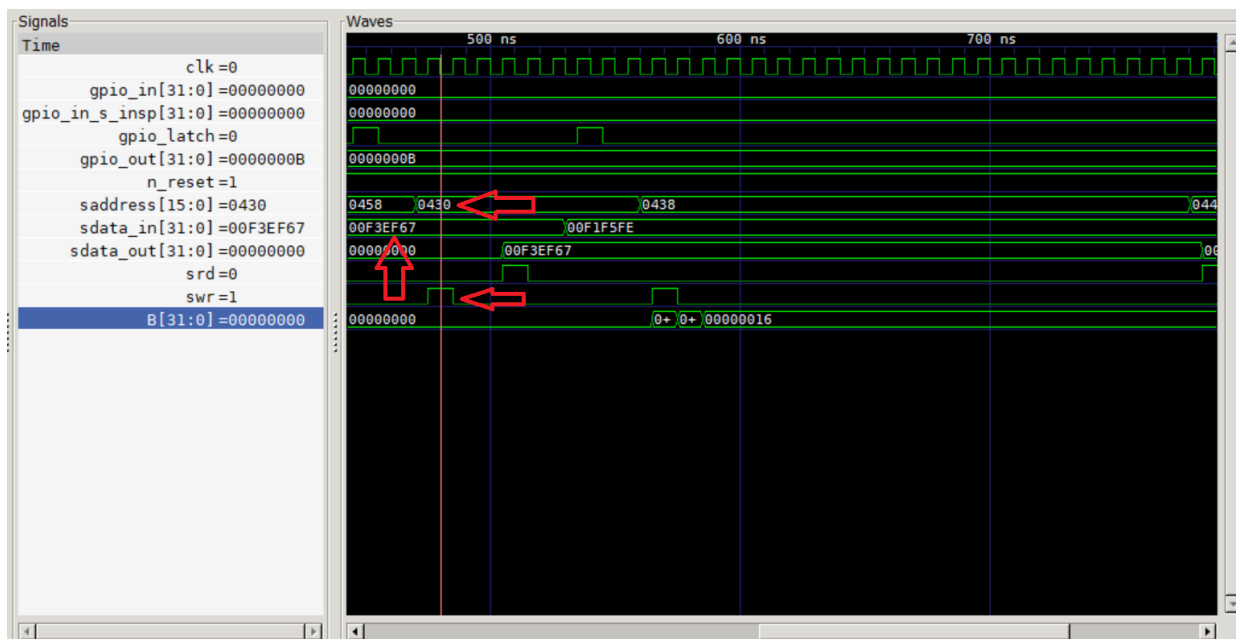
Rys.1 Wynik mnożenia liczb 0x2137 oraz 0x125 w zapisie heksadecymalnym

Na powyższym zrzucie ekranu zaznaczono cyframi adresy, do których się odwoływano w trakcie symulacji, a strzałkami wartości zapisane na magistralę lub z niej odczytane.

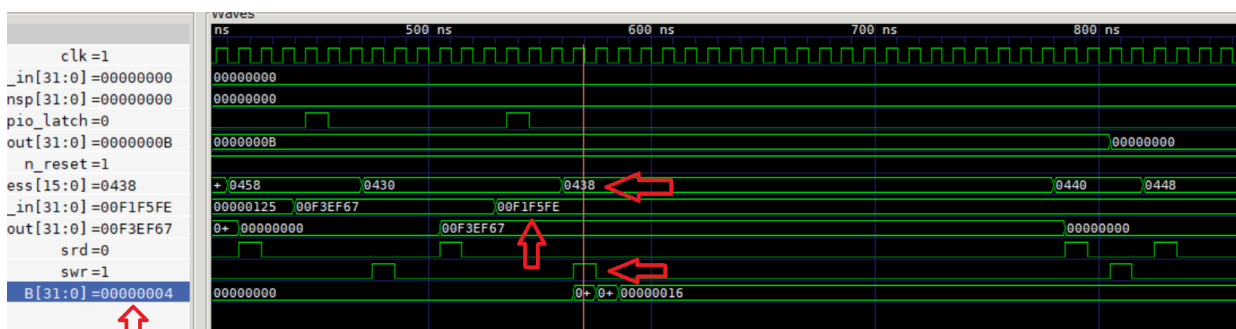
1. Adres rejestru A1, na który zapisano wartość przedstawioną w zapisie heksadecymalnym 0x2137.
2. Adres rejestru A2, na który zapisano wartość przedstawioną w zapisie heksadecymalnym 0x125.
3. Odczyt spod adresu rejestru B, w celu sprawdzenia statusu. Wartość statusu to 0x0 co oznacza, że operacja mnożenia została wykonana.
4. Odczyt spod adresu rejestru W w celu otrzymania wyniku. Otrzymana wartość to 0x2603F3 w zapisie heksadecymalnym. Jest to prawidłowy wynik mnożenia liczb 0x2137 oraz 0x125.
5. Zapis na gpio_out spod adresu rejestru L odpowiedzialnego za przetrzymywanie liczby jedynek w wyniku. W tym przypadku jest to 0xB policzonych jedynek co jest prawidłowym wynikiem.

3.2 Test 2: Wynik nie mieści się na 32 bitach

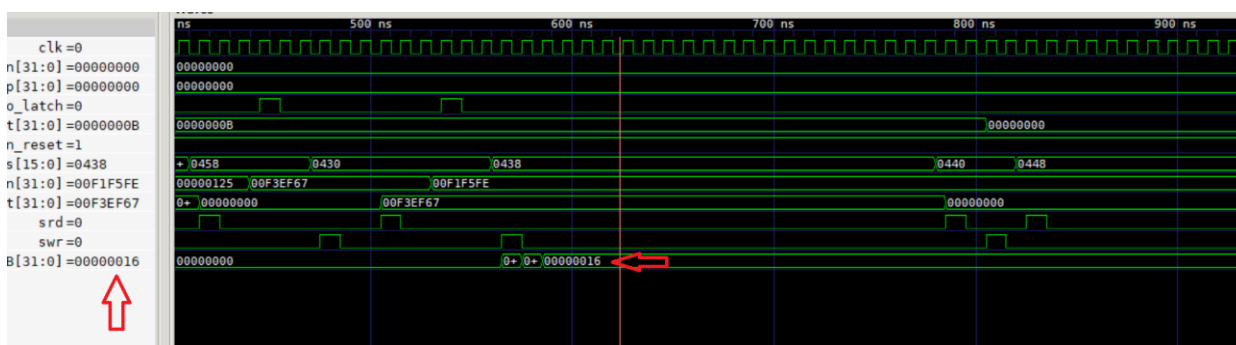
Rejestr B odpowiada za przechowywanie wartości opisującej stan modułu. Jeżeli wynik operacji mnożenia nie mieści się na 32 bitach wtedy rejestr statusu przyjmuje wartość 0x16 i w ten sposób powiadamia o zaistniałym błędzie. Poniżej znajdują się zrzuty ekranu prezentujące taki scenariusz zasymulowany w narzędziu gtkwave.



Rys.3 Zapis wartości 0x00F3EF67 do rejestru A1



Rys.4 Zapis wartości 00F1F5FE do rejestru A2, rejestr B przyjmuje wartość 0x4 - rozpoczęcie operacji mnożenia



Rys.5 Zakończenie mnożenia - zidentyfikowanie nie mieszczącego się wyniku na 32 bitach – rejestr statusu B przyjmuje wartość 0x16

4.Testowanie systemu

4.1 Uruchomienie systemu operacyjnego oraz dodanie modułu do jądra Linux

```
sykt@deb4sykom04:~/temp$ ./skrypt.sh
qemu-system-riscv32-sykt: info: Qemu for SYKT lecture made by A.Pruszkowski (Compiled at: Feb 23 2023 13:00:53)
Qemu internal configuration was found.
QEMU start report:
Compilation: Feb 23 2023 13:00:52
NIC: enp0s3
NIC MAC: 08:00:2f:bb:ea:d1 (17)
HASH: 0x7802d52b39e99751
NIC2: 0x000008002fbb:ea:d1
Qemu internal configuration was found.
QEMU start report:
Compilation: Feb 23 2023 13:00:52
NIC: enp0s3
NIC MAC: 08:00:2f:bb:ea:d1 (17)
HASH: 0x7802d52b39e99751
NIC2: 0x000008002fbb:ea:d1
GPIO Emulator for QEMU initializing .... (Compiled at: Jun 2 2023 18:06:46)
qemu-system-riscv32-sykt: info: Qemu for SYKT lecture begin (SYKT_IO)

OpenSBI v0.5 (Mar 17 2021 16:41:14)

Platform Name       : QEMU Virt Machine
Platform HART Features : RV32ACDFIMSU
Platform Max HARTs   : 8
Current Hart        : 0
Firmware Base       : 0x80000000
Firmware Size       : 124 KB
Runtime SBI Version  : 0.2
```

Rys.7 Uruchomienie nowo utworzonego systemu operacyjnego skryptem `.sh`.

```
# cat /proc/cpuinfo
processor       : 0
hart          : 0
isa           : rv32imafdcu

# █
```

Rys.8 Weryfikacja działania systemu operacyjnego komendą „`cat /proc/cpuinfo`”

```
Welcome to Buildroot
buildroot login: root
# modprobe kernel_module
[ 14.323444] Starting SYKT module ...
# █
```

Rys.9 Dodanie modułu do jądra

```
# rmmod kernel_module
[ 30.881898] Cleaning up SYKT module.
qemu-system-riscv32-sykt: info: sykt_test_write(addr=0x0, val64=0x7f3333, size=0x4) - FINISHER_FAIL
qemu-system-riscv32-sykt: info: Qemu for SYKT lecture finish (5668)
Erase is control-H (^H).
sykt@deb4sykom10:~/temp$
```

Rys.10 Usunięcie modułu z jądra oraz zakończenie pracy systemu operacyjnego

Przeprowadzono dwa rodzaje testów w celu weryfikacji poprawności działania systemu. Pierwszy test polegał na zapisie wartości oraz odczycie wyników z konsoli. Do drugiego testu użyto aplikacji testującej poprawność działania modułu.

4.2 Testy w konsoli

```
# echo "19" > /sys/kernel/sykt/sjda1
# echo "27" > /sys/kernel/sykt/sjda2
# cat < /sys/kernel/sykt/sjdw
0x3cf
# cat < /sys/kernel/sykt/sjdl
0x8
# cat < /sys/kernel/sykt/sjdb
0x0
# cat < /sys/kernel/sykt/sjda1
0x19
# cat < /sys/kernel/sykt/sjda2
0x27
#
```

Rys.11 Pierwszy test systemu w konsoli

```
# echo "211" > /sys/kernel/sykt/sjda1
# echo "123" > /sys/kernel/sykt/sjda2
# cat < /sys/kernel/sykt/sjdw
0x25953
# cat < /sys/kernel/sykt/sjdl
0x9
# cat < /sys/kernel/sykt/sjdb
0x0
# cat < /sys/kernel/sykt/sjda1
0x211
# cat < /sys/kernel/sykt/sjda2
0x123
#
```

Rys.12 Drugi test systemu w konsoli

```
# echo "2222222" > /sys/kernel/sykt/sjda1
# echo "1333" > /sys/kernel/sykt/sjda2
# cat /sys/kernel/sykt/sjdw
0x0
# cat /sys/kernel/sykt/sjdl
0x0
# cat /sys/kernel/sykt/sjdb
0x16
# cat /sys/kernel/sykt/sjda1
0x222222
# cat /sys/kernel/sykt/sjda2
0x1333
#
```

Rys.13 Trzeci test systemu w konsoli – wynik nie mieści się na 32 bitach

Zadane wartości system przyjął jako wartości heksadecymalne, stąd wyniki są wartościami otrzymanymi z mnożenia dwóch liczb w zapisie heksadecymalnym. Kolejne wartości otrzymane komendą „cat” w trakcie każdego testu to:

- Wynik mnożenia w zapisie heksadecymalnym.
- Liczba jedynek zliczonych w wyniku
- Wartość rejestru B – 0x0 oznacza, że system jest gotowy do następnej operacji mnożenia. Wartość rejestru 0x16 oznacza, że wynik nie mieści się na 32 bitach.
- Wartość rejestru A1
- Wartość rejestru A2

W trzecim scenariuszu wynik nie mieści się na 32 bitach o czym informuje rejestr B. Z tego powodu system wypisał zera w wyniku oraz w zwracanej wartości liczby jedynek.

4.3 Testy wykonane aplikacją użytkownika

Poniżej umieszczono zrzut ekranu przedstawiający wyniki testów wykonanych aplikacją użytkownika:

```
# ./main
[ 13.678929] random: fast init done
Compiled at Jun  6 2023 12:25:19
Wynik mnożenia liczb 2 i 4 wynosi: 0x8
Liczba jedynek w wyniku wynosi: 0x1
Rejestr statusu: 0x0
Rejestr A1: 0x2
Rejestr A2: 0x4

Wynik mnożenia liczb 19 i 27 wynosi: 0x3cf
Liczba jedynek w wyniku wynosi: 0x8
Rejestr statusu: 0x0
Rejestr A1: 0x19
Rejestr A2: 0x27

Wynik mnożenia liczb 211 i 123 wynosi: 0x25953
Liczba jedynek w wyniku wynosi: 0x9
Rejestr statusu: 0x0
Rejestr A1: 0x211
Rejestr A2: 0x123

Wynik mnożenia liczb 2222 i 1333 wynosi: 0x28f52c6
Liczba jedynek w wyniku wynosi: 0xd
Rejestr statusu: 0x0
Rejestr A1: 0x2222
Rejestr A2: 0x1333

Wynik mnożenia liczb 2222222 i 1333 wynosi: 0x0
Liczba jedynek w wyniku wynosi: 0x0
Rejestr statusu: 0x16
Rejestr A1: 0x2222222
Rejestr A2: 0x1333
```

Rys.14 Test aplikacją użytkownika

Na zrzucie ekranu widać kolejne wyniki testów systemu. Scenariusz ostatniego testu zakłada, że otrzymana liczba nie mieści się na 32 bitach stąd 0x0 w wyniku oraz liczbie zliczonych jedynek. Wartość rejestru statusu wynosi 0x16 co informuje o nie zmieszczeniu się wyniku.