

Технически университет – Габрово
Катедра: “Компютърни Системи и Технологии”

КУРСОВ ПРОЕКТ

по дисциплина
“Компютърни Мрежи”

*Разработил: Ивайло Тихомиров Белев
№20405114, КСТ 2А, IV курс*

Габрово 2009

Проверил:
(доц. Р. Райчев)

Задача

Да се реализира клиент-сървърно мрежово приложение за два броя Spartan-3 FPGA Starter Kit Board.

Реализираното приложение е играта „Бикове и крави“.

I. Същност на играта

1. Цел

„Бикове и крави“ в „човешкия“ ѝ вариант е предназначена за двама играчи. Всеки един от тях си намисля и записва едно 4-цифрено число, така че противникът му да не може да го види (включително и по време на игра). Целта е да се познае числото на другия играч.

2. Правила

- Числото се състои от 4 неповтарящи се цифри, като не може да започва с 0.
- Играчите се редуват при изказването на своите предположения за числото на другия.
- „Играч 1“ започва първи. „Играч 2“ има право на последен ход. Ако „Играч 1“ познае числото на „Играч 2“, а при своя последен ход „Играч 2“ познае числото на „Играч 1“, тогава играта завършва наравно.
- След като единият играч е изказал предположение, другият играч трябва да каже, доколко е вярно предположението. Ако има цифра от предположеното число в числото на противника – това е „крава“ (○). Ако цифрата е и на същата позиция – това е „бик“ (⊗).

Пример

Играч 1: |2085|

Предположение/Отговор:

9204 ○○
7845 ⊗○

...

Играч 2: |7982|

Предположение/Отговор:

5349 ○
8056 ⊗○○

...

Добрите играчи познават числото на противника за 7-8 хода.

3. Реализиран вариант

Единият Spartan кит служи за сървър, а другият е клиент. Сървърът генерира число, а потребителят посредством клиента трябва да го познае, като се стреми това да стане за минимален брой ходове.

II. Хардуерно описание

1. Необходим хардуер

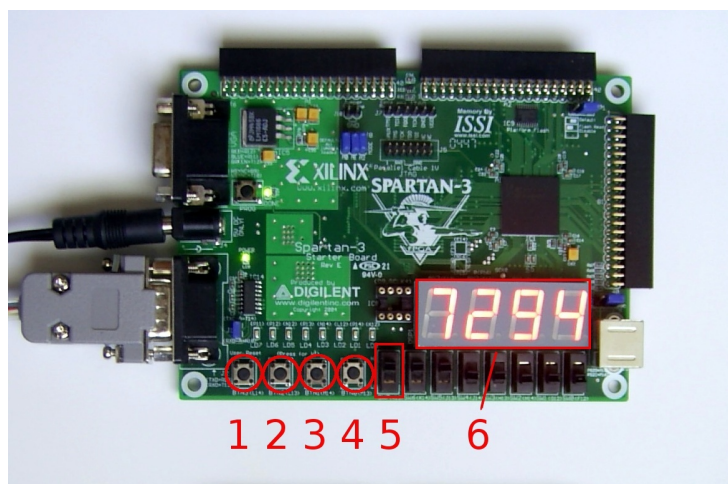
За реализацията на този проект са необходими два Spartan-3 FPGA Starter Kit Board. За осъществяване на комуникацията между тях се използва интерфейса RS232. Кабелът, който се използва е кръстосан, като свързването на пиновете е следното: TXD1->RXD2; RXD1->TXD2; GROUND1->GROUND2.

2. Сървър

Сървърът генерира псевдослучайно число, което може да бъде показано на 4-цифровия, 7-сегментен дисплей. Показанието се включва и изключва с най-левия суич (SW7).

3. Клиент

Посредством клиента потребителят избира и изпраща предполагаемото число. Текущо избраното число се показва на 4-цифровия, 7-сегментен дисплей (фиг. 1-6).



Фиг.1: Игрови средства

Функции на бутоните:

BTN0 (фиг.1-4) – увеличаване/намаляване на текущо избраната цифра

BTN1 (фиг.1-3) – избиране на следващата/предходната цифра

BTN2 (фиг.1-2) – изпращане на числото/изчистване на отговора от сървъра

BTN3 (фиг.1-1) – рестартиране

SW7 (фиг.1-5) – определя поведението на BTN0 и BTN1; инкрементиране/декрементиране, движение наляво/надясно

След изпращане на числото, потребителят получава отговор от сървъра, доколко има съвпадение.

Възможни отговори (фиг. 2):

„-“ - числото не отговаря на правилата на играта

„_“ - няма съвпадение

„0“ - крава

„X“ - бик

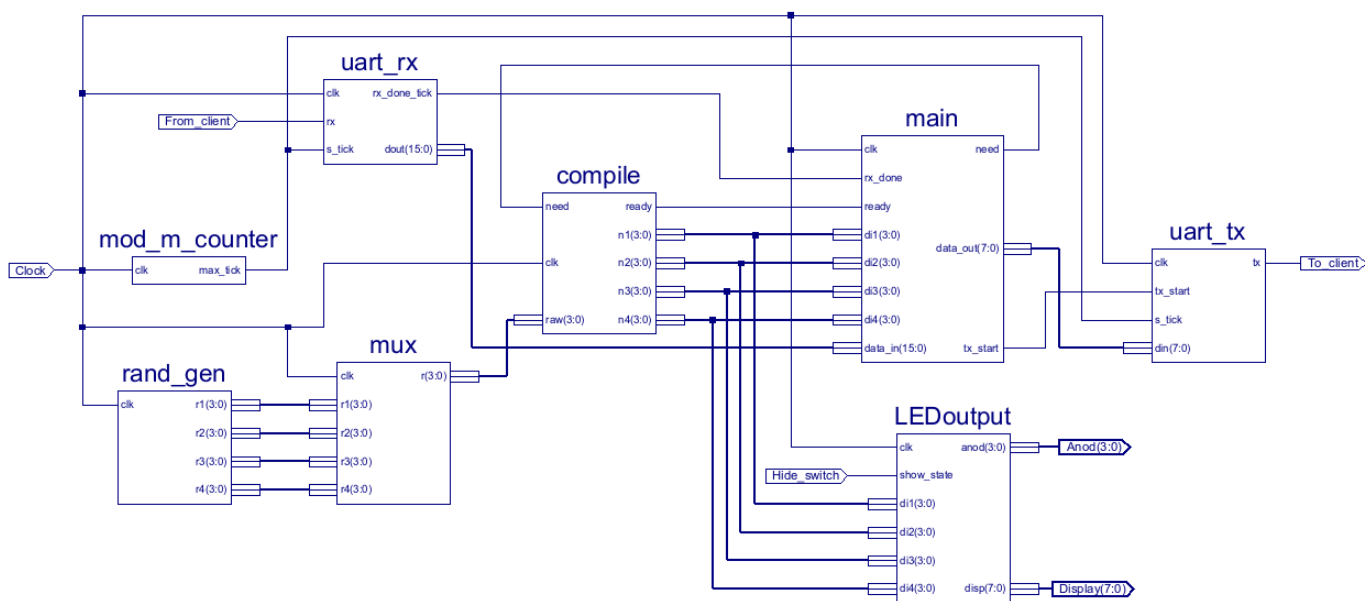


Фиг. 2: Възможни отговори

III. Софтуерна реализация

Проектът е реализиран на VHDL, а средата за разработка е Xilinx ISE WebPACK 10.1 за Linux (работено е под Ubuntu 8.10).

1. Сървър (фиг. 3)



Фиг. 3: Схема на сървъра

а) Основни блокове:

- Главен блок (main)
- Блок за генериране на число (rand_gen, mux, compile)
- Блок за извеждане на генерираното число (ledoutput)
- Блок за вход (mod_m_counter, uart_rx)
- Блок за изход (mod_m_counter, uart_tx)

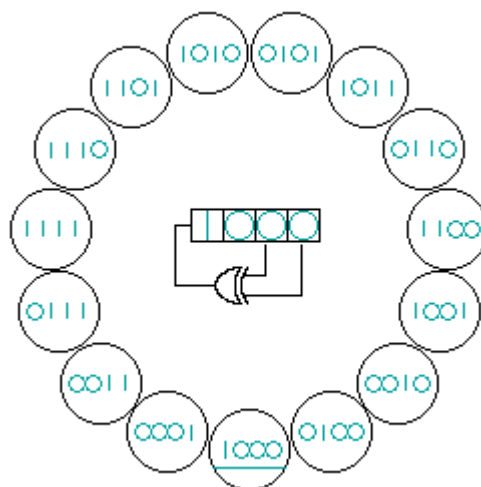
б) Главен блок

Служи за връзка и координация между всички други блокове.

При сигнали включване / рестарт / нова игра, подадени от клиента, от блока за вход, се изпраща (едва тогава!) заявка за генериране на число. Така се заобикаля проблема с генерирането на една и съща поредица псевдослучайни числа при всяко включване на китовите. След получаване на генерираното число се прави сравнение с предполагаемото число (от блока за вход) и резултатът се изпраща към блока за изход.

в) Блок за генериране на число

Състои се от 4 4-битови преместващи регистри (фиг. 4) с обратна връзка, които на всеки такт преместват по един бит. Така след 4 такта всички битове на всеки от регистрите са изцяло подменени. Мултиплексорът на всеки такт подава число от един от преместващите



Фиг. 4: 4-битов преместващ регистър с обратна връзка и всичките му възможни състояния

регистри на следващия блок, който от своя страна „сглобява“ валидно за играта число, и го подава на главния блок.

г) Блок за извеждане на генерираното число

Извежда генерираното за текущата игра число на 4-цифровия, 7-сегментен дисплей.

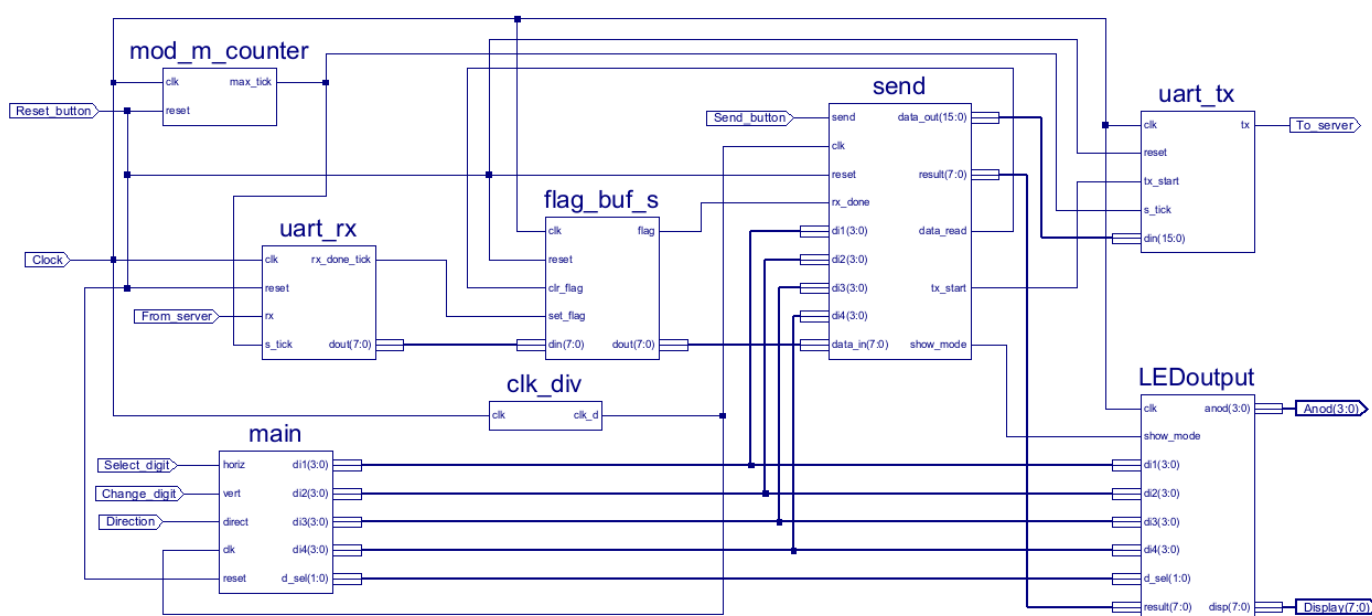
д) Блок за вход

Приема данните от клиента. Използва се протокол за връзка с 1 старт бит, 16 бита данни, 1 стоп бит. Брояч определя средата на всеки бит, и тогава е момента на четенето. Данните се подават на главния блок.

е) Блок за изход

Изпраща отговор на клиента. Използва се протокол за връзка с 1 старт бит, 8 бита данни, 1 стоп бит. Брояч определя скоростта на подаване на битовете. Данните се подават на главния блок.

2. Клиент (фиг. 5)



Фиг. 5: Схема на клиента

а) Основни блокове:

- Главен блок (main, clk_div)
- Блок за изпращане на данни (send, clk_div)
- Блок за извеждане на текущо състояние (ledoutput)
- Блок за вход (mod_m_counter, uart_rx, flag_buf_s)
- Блок за изход (mod_m_counter, uart_tx)

б) Главен блок

Служи за избиране на предполагаемото число. То се подава към блока за изпращане и към дисплея.

в) Блок за изпращане на данни

При заявка от потребителя за изпращане на текущото число, проверява неговата валидност и евентуално го подава на блока за изход. След това приема отговора от сървъра и го подава на блока за извеждане върху дисплея.

г) Блок за извеждане на текущо състояние

Извежда текущото число и отговора от сървъра на 4-цифровия, 7-сегментен дисплей.

д) Блок за вход

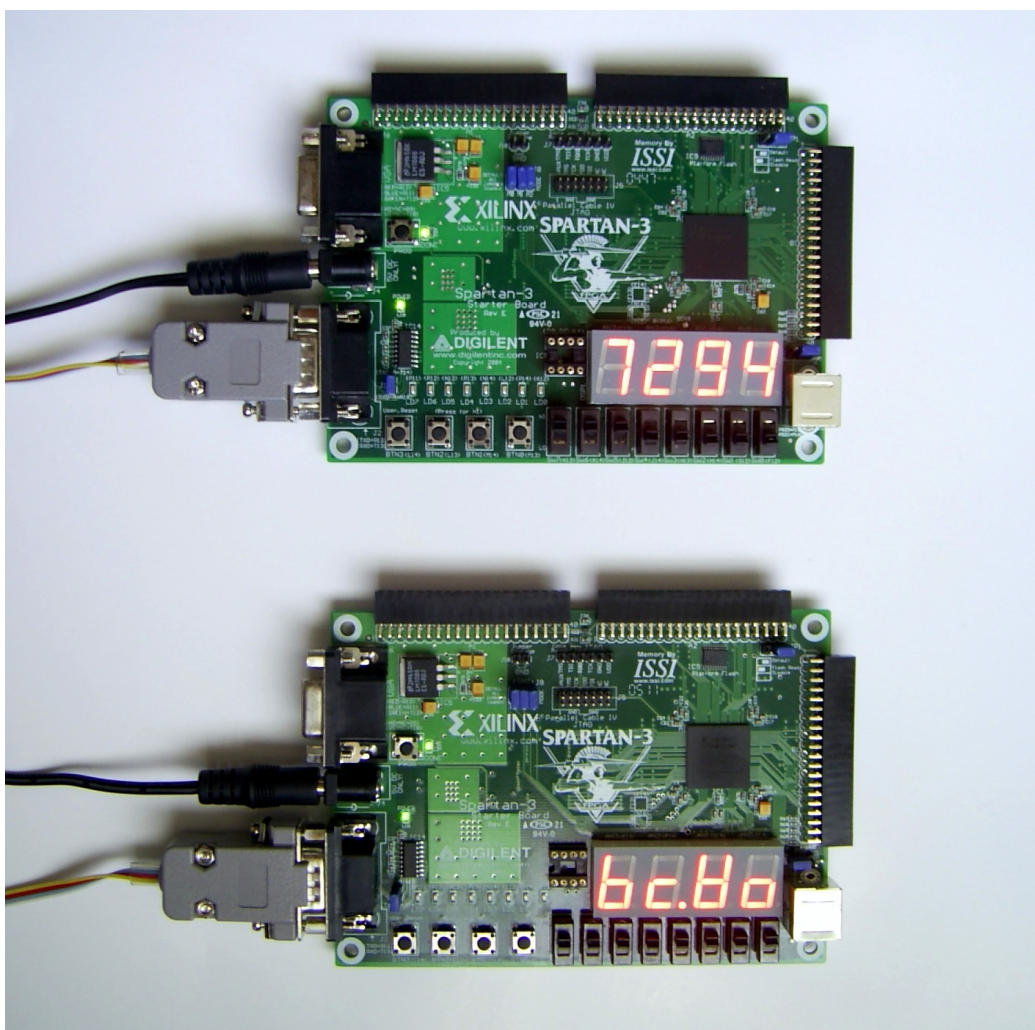
Приема отговор от сървъра. Използва се протокол за връзка с 1 старт бит, 8 бита данни, 1 стоп бит. Брояч определя средата на всеки бит, и тогава е момента на четенето. Данните се подават на блока за изпращане. Има буфер, който служи за връзка на блоковете, тъй като работят на различни честоти.

е) Блок за изход

Изпраща запитване до сървъра. Използва се протокол за връзка с 1 старт бит, 16 бита данни, 1 стоп бит. Брояч определя скоростта на подаване на битовете. Данните се подават на блока за изпращане.

IV. Заключение

Реализираното приложение (фиг. 6), освен че онагледява част от възможностите на Spartan-3 FPGA Starter Kit Board, но и след това остава занимателно за потребителя неограничено дълго време. Играта предизвиква потребителя да тества своето логическо мислене и посредством практика да го развива.



Фиг. 6: Играта в действие

Списък на файловете

1. Bulls & Cows. pdf – този файл:)
2. Bulls & Cows.htm – описание на проекта
3. Bulls & Cows - whole project.7z – целият проект (архивиран)
 - BCCClient.ise – проектен файл
 - assembled.sch – цялостна схема на свързването на отделните модули
 - ports.ucf – връзка към физическите пинове
 - *.vhd – VHDL сорс-кодове
 - ...
4. Bulls & Cows - for programming.7z – готови файлове за директно програмиране на платките (архивирани)
 - *.bit – bitstream файл
 - *.mcs – PROM файл
5. BCCClient_scheme.png – схема на клиента
6. BCServer_scheme.png – схема на сървъра
7. Spartan_single.jpg – използвани елементи от платката
8. Spartan_married.jpg – играта в действие
9. shift_reg.png – преместващ регистър
10. nothing_cow_bull.jpg – възможни отговори от сървъра
11. email.png – е-мейл на автора на проекта

Литература

1. Spartan-3 FPGA Starter Kit Board User Guide (pdf)
2. VHDL-Handbook (pdf)
3. FPGA Prototyping by VHDL Examples (pdf)
4. Въведение в VHDL (pdf)
5. Linear feedback shift register:
<http://en.wikipedia.org/wiki/LFSR>
6. Random Number Generation in VHDL:
www.mydatabasesupport.com/forums/arch/365312-random-number-generation-vhdl.html