



Министерство науки и высшего образования Российской Федерации
Федеральное государственное бюджетное образовательное учреждение
высшего образования
«Московский государственный технический университет имени
Н.Э. Баумана
(национальный исследовательский университет)»
(МГТУ им. Н.Э. Баумана)

ФАКУЛЬТЕТ «Информатика и системы управления»

КАФЕДРА «Программное обеспечение ЭВМ и информационные технологии»

Отчет по лабораторной работе №1 по дисциплине "Архитектура ЭВМ"

Тема Проектирование систем на кристалле на основе ПЛИС

Студент Костев Д.

Группа ИУ7-51Б

Оценка (баллы) _____

Преподаватель _____

Оглавление

1	Цели лабораторной работы:	2
2	Аналитическая часть	3
2.1	Функциональная схема разрабатываемой системы на кристалле.	3
3	Практическая часть	5
3.1	Модуль в QSYS	5
3.2	Создание проекта Nios2	7
4	Вывод	9
	Литература	10

1 | Цели лабораторной работы:

Цель работы: изучение основ построения микропроцессорных систем на ПЛИС. В ходе работы необходимо ознакомиться с принципами построения систем на кристалле (СНК) на основе ПЛИС, получить навыки проектирования СНК в САПР Altera Quartus II, выполнить проектирование и верификацию системы с использованием отладочного комплекта Altera DE1Board.

2 | Аналитическая часть

2.1 Функциональная схема разрабатываемой системы на кристалле.

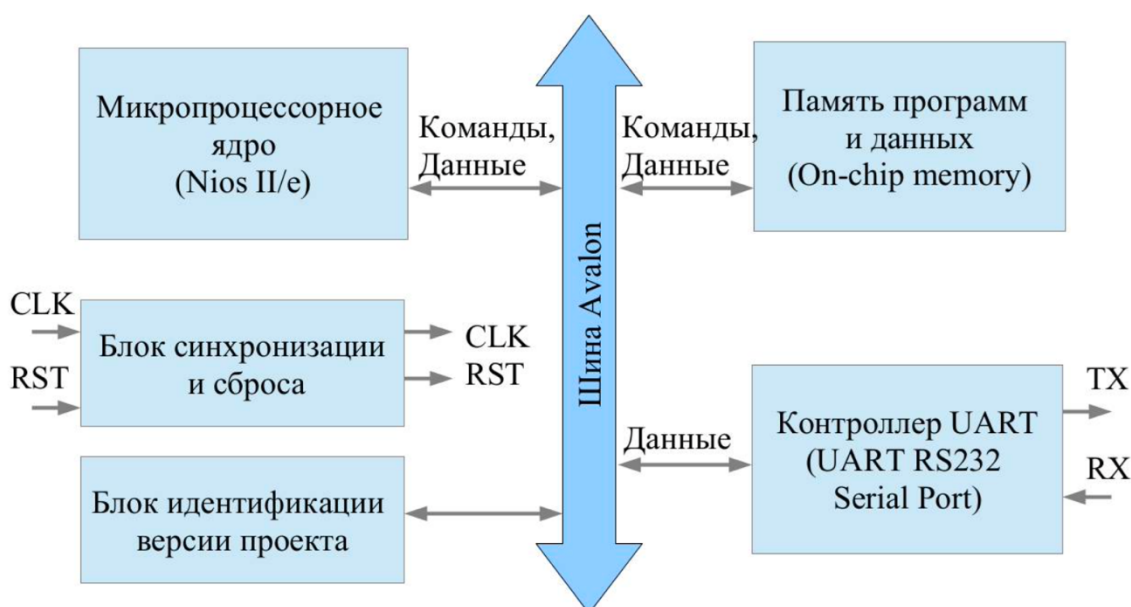


Рис. 2.1: Функциональная схема разрабатываемой системы на кристалле

Система на кристалле состоит из следующих блоков:

- Микропроцессорное ядро Nios II/e выполняет функции управления системой.
- Внутренняя оперативная память СНК, используемая для хранения программы управления и данных.
- Системная шина Avalon обеспечивает связность всех компонентов системы.
- Блок синхронизации и сброса обеспечивает обработку входных сигналов сброса и синхронизации и распределение их в системе. Внутренний сигнал сброса синхронизирован и имеет необходимую для системы длительность.
- Блок идентификации версии проекта обеспечивает хранение и выдачу уникального идентификатора версии, который используется программой управления при инициализации системы.

- Контроллер UART обеспечивает прием и передачу информации по интерфейсу RS232.

3 | Практическая часть

3.1 Модуль в QSYS

1. Был создан новый модуль Qsys.
2. Установлена частота внешнего сигнала синхронизации 50 000 000 Гц.
3. Добавлен в проект модуль синхронизируемого микропроцессорного ядра Nios2.
4. Добавлен в проект модуль ОЗУ программ и данных.
5. Добавлены компоненты Avalon System ID, Avalon UART.
6. Создана сеть синхронизации и сброса системы.
7. Сигналы TX и RX экспортированы во внешние порты.
8. Назначены базовые адреса устройств.

Итог выполненных действий показан на рисунке 3.1. На рисунке 3.2 показана таблица распределения адресов.

uments/kate(nios.qsys)


System Contents										Address Map	Clock Settings	Project Settings	Instance Parameters	System Inspector	HDL Example	Generation
Use	Connections	Name	Description	Export	Clock	Base	End	IRQ	Opcode Name							
<input checked="" type="checkbox"/>		clk_0	Clock Source	clk reset Double-click to export Double-click to export	clk_0											
<input checked="" type="checkbox"/>		clk_in	Clock Input													
<input checked="" type="checkbox"/>		clk_in_reset	Reset Input													
<input checked="" type="checkbox"/>		clk_reset	Clock Output													
<input checked="" type="checkbox"/>		clk_reset	Reset Output													
<input checked="" type="checkbox"/>		nios2_qsys_0	Nios II Processor													
<input checked="" type="checkbox"/>		clk	Clock Input	Double-click to export	clk_0											
<input checked="" type="checkbox"/>		reset_n	Reset Input	Double-click to export	[clk]											
<input checked="" type="checkbox"/>		data_master	Avalon Memory Mapped Master	Double-click to export	[clk]			IRQ 0	IRQ 31							
<input checked="" type="checkbox"/>		instruction_master	Avalon Memory Mapped Master	Double-click to export	[clk]											
<input checked="" type="checkbox"/>	jtag_debug_module_re	Reset Output	Double-click to export	[clk]												
<input checked="" type="checkbox"/>	jtag_debug_module	Avalon Memory Mapped Slave	Double-click to export	[clk]	# 0x8800		0x8fff									
<input checked="" type="checkbox"/>	custom_instruction_m	Custom Instruction Master	Double-click to export	[clk]												
<input checked="" type="checkbox"/>	onchip_memory2_0	On-Chip Memory (RAM or ROM)														
<input checked="" type="checkbox"/>	clk1	Clock Input	Double-click to export	clk_0												
<input checked="" type="checkbox"/>	s1	Avalon Memory Mapped Slave	Double-click to export	[clk1]	# 0x4000		0x4003									
<input checked="" type="checkbox"/>	reset1	Reset Input	Double-click to export	[clk1]												
<input checked="" type="checkbox"/>	sysid_qsys_0	System ID Peripheral														
<input checked="" type="checkbox"/>	clk	Clock Input	Double-click to export	clk_0												
<input checked="" type="checkbox"/>	reset	Reset Input	Double-click to export	[clk]												
<input checked="" type="checkbox"/>	control_slave	Avalon Memory Mapped Slave	Double-click to export	[clk]	# 0x9048		0x904f									
<input checked="" type="checkbox"/>	uart_0	UART (RS-232 Serial Port)														
<input checked="" type="checkbox"/>	clk	Clock Input	Double-click to export	clk_0												
<input checked="" type="checkbox"/>	reset	Reset Input	Double-click to export	[clk]												
<input checked="" type="checkbox"/>	s1	Avalon Memory Mapped Slave	Double-click to export	[clk]	# 0x9020		0x903f									
<input checked="" type="checkbox"/>	external_connection	Conduit	uart0													

Рис. 3.1: Модуль QSYS

System Contents	Address Map	Clock Settings	Project Settings	Instance Parameters	System Inspector	HDL Example	Generation
		nios2_qsys_0.data_master		nios2_qsys_0.instruction_master			
nios2_qsys_0.jtag_debug_module	0x8800 - 0x8fff					0x8800 - 0x8fff	
onchip_memory2_0.s1	0x4000 - 0x6003					0x4000 - 0x6003	
sysid_qsys_0.control_slave	0x9048 - 0x904f					0x9048 - 0x904f	
uart_0.s1	0x9020 - 0x903f					0x9020 - 0x903f	

Рис. 3.2: Таблица распределения адресов

3.2 Создание проекта Nios2

В файл helloworldsmall.c был добавлен код эхо-программы приема-передачи по интерфейсу RS232, представленный на листинге 3.1. Также был создан образ ОС HAL с драйверами устройств, используемых в аппаратном проекте.

Листинг 3.1: Функция нахождения расстояния Левенштейна рекурсивно

```
1 #include "sys/alt_stdio.h"
2
3 int main()
4 {
5     char ch;
6     alt_putstr("Hello from System on Chip\n");
7     alt_putstr("Send any character\n");
8     /* Event loop never exits. */
9     while (1) {
10         ch=alt_getchar();
11         alt_putchar(ch);
12     }
13     return 0;
14 }
```

После успешной сборки и выполнения код программы был доработан: были добавлены строки, передающие по UART значение SystemID в виде четырех байт символов в ASCII формате.. Результат доработки представлен на листинге 3.2.

Листинг 3.2: Функция нахождения расстояния Левенштейна рекурсивно

```
1 #include "sys/alt_stdio.h"
2 #include "system.h"
3 #include "altera_avalon_sysid_qsys.h"
4 #include "altera_avalon_sysid_qsys_regs.h"
5
6 int main()
7 {
8     char ch;
9     alt_putstr("Hello from System on Chip\n");
10    alt_putstr("Send any character\n");
11    int id = IORD_ALTERA_AVALON_SYSID_QSYS_ID(SYSID_QSYS_0_BASE);
12    char a[6];
13    int i = 1;
14    while (id)
15    {
16        a[4 - i] = '0' + id % 10;
17        id /= 10;
18        i++;
19    }
20    a[4] = '\n';
21    a[5] = '\0';
22    for (int i = 0; i < 5; i++)
23        alt_putchar(a[i]);
```



```

24  /* Event loop never exits. */
25  while (1)
26  {
27      ch=alt_getchar();
28      alt_putchar(ch);
29  }
30  return 0;
31  }

```

Результат выполнения программы показан на рисунке 3.3. **Замечание:** программа не собралась на моем компьютере из-за отсутствия драйверов, в моем же варианте было бы число 5109.

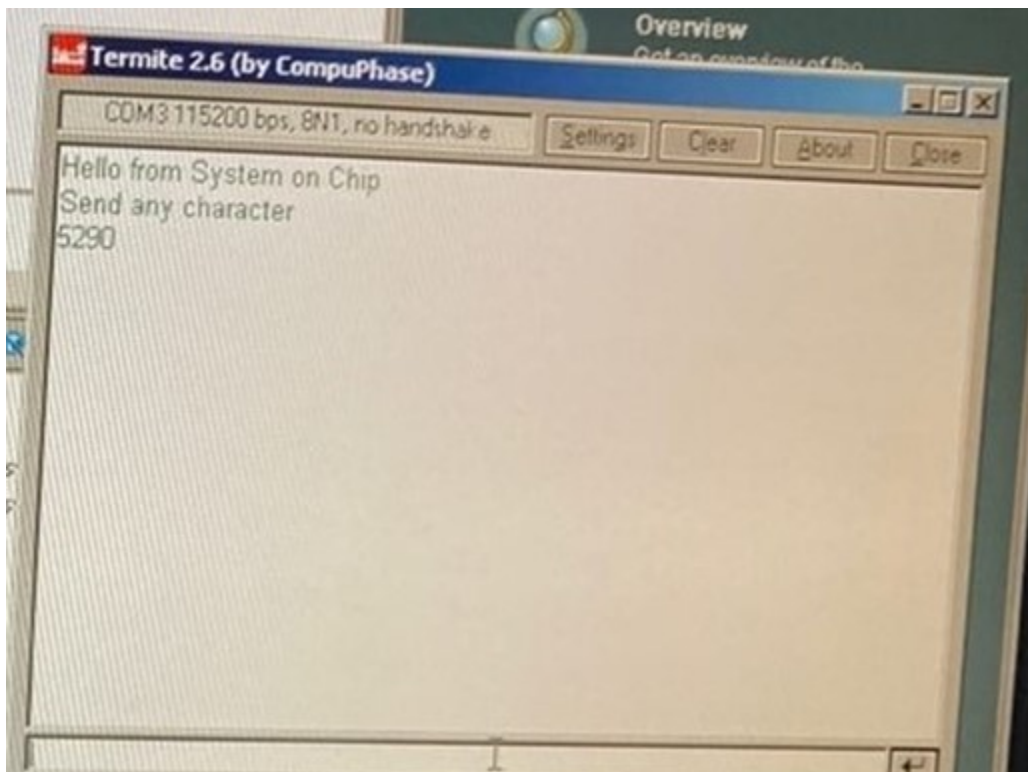


Рис. 3.3: Результат выполнения программы

4 | Вывод

В ходе данной лабораторной работы были изучены основы построения микропроцессорных систем на ПЛИС, получены навыки проектирования СНК в САПР Altera Quartus II, также были выполнены проектирование и верификация системы с использованием отладочного комплекта Altera DE1Board.

Литература

1. Методические указания к ЛР1 по ЭВМ. URL: <https://e-learning.bmstu.ru/iu6/pluginfile.php/16762/nresource/content/2/'ЭВМ-ЛР-Разработка-СнК.pdf>, 01.10.2021