

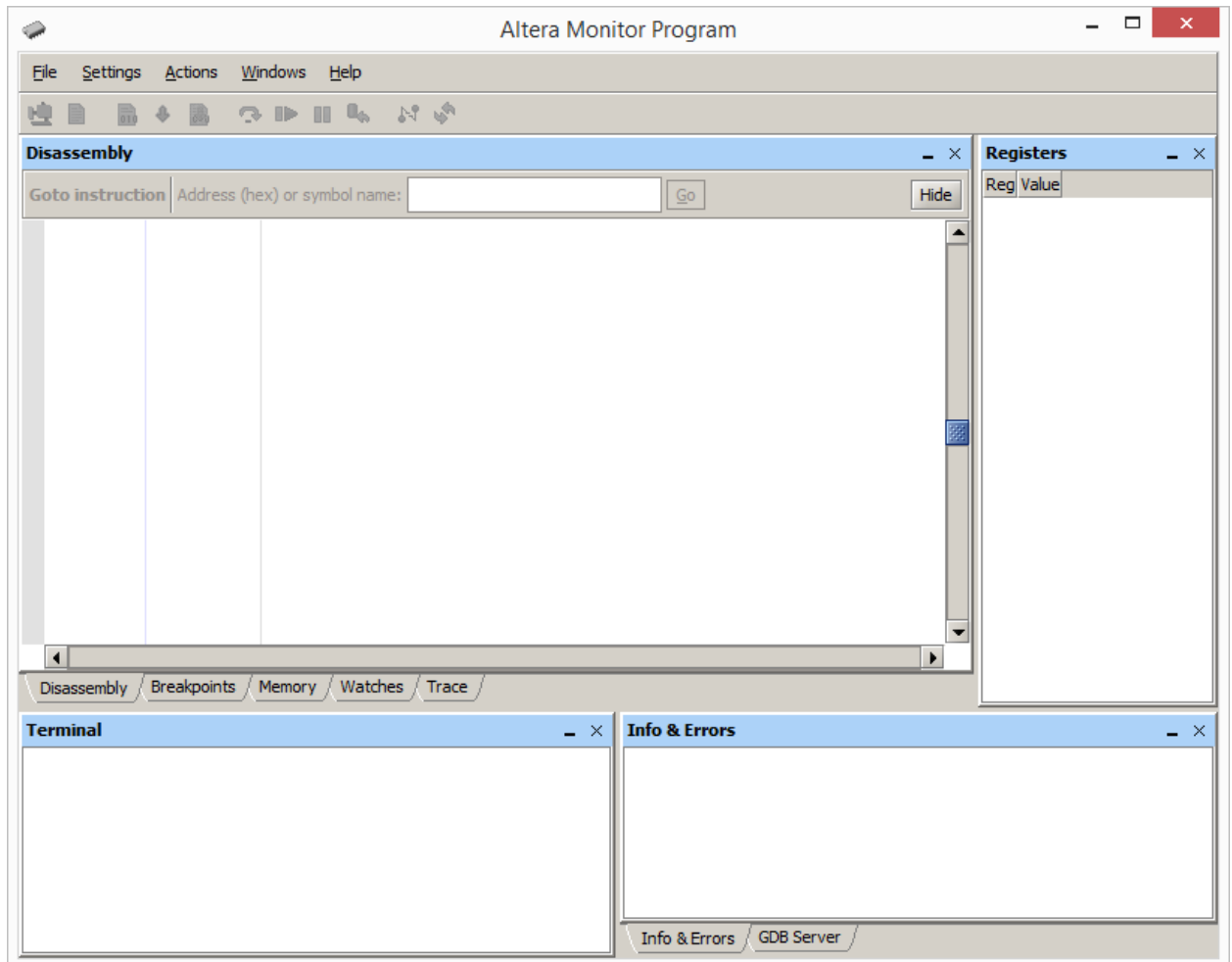
Лабораторная работа № 1

Обращение к памяти при операциях ввода-вывода.

Работа выполняется в пакете Altera Monitor Program. В ходе работы исследуется выполнение тестовой программы, позволяющей производить ввод и вывод, используя стандартную периферию макета DE1-SoC.

Порядок выполнения работы.

1. Подключить макет DE1-SoC.
2. Открыть Altera Monitor Program, следуя по пути: Файловая система – opt – altera15.0 – University Program – Monitor Program – bin – altera monitor program.
3. В появившемся окне выбрать File > New Project.



4. Далее выбрать имя папки, определить имя проекта (папка Exercise1, проект lab1). Выбрать архитектуру ARM Cortex-A9. Клик Next

New Project Wizard

Specify a project name and directory

Project directory:
U:\Exercise1\Part1

Project name:
lab1_part1

Architecture:

5. Выбрать DE1-SoC Computer, Next.

New Project Wizard

Specify a system

Select a system

DE1-SoC Computer Documentation

This system, called the DE1-SoC Computer, is intended to be used as a platform for experiments in computer organization and embedded systems. To support these experiments, the system contains a number of components: a processor, memory, audio and video devices, and some simple I/O peripherals.

System details

System description file (SOPCInfo):
ersity_Program/Computer_Systems/DE1-SoC/DE1-SoC_Computer/verilog/Computer_System.sopcinfo Browse...

Quartus II programming (SOF) file (optional):
niversity_Program/Computer_Systems/DE1-SoC/DE1-SoC_Computer/verilog/DE1_SoC_Computer.sof Browse...

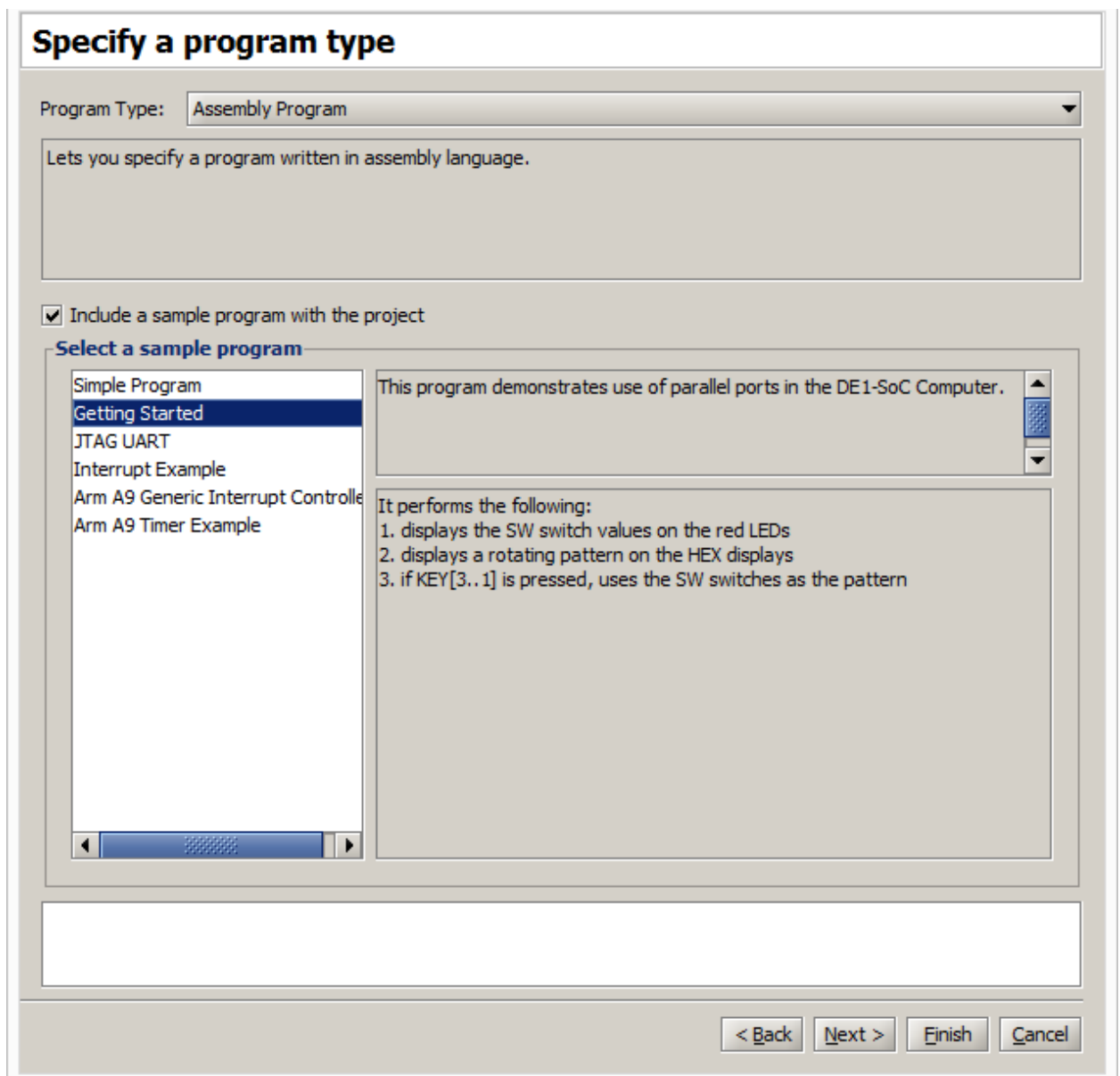
The SOF file represents the FPGA programming file for the hardware system. If it is specified here, then the Monitor Program can be used to download this programming file onto the board. Otherwise, the system will need to be downloaded using some other method (for example, by using Quartus II).

Quartus II JTAG debugging information (JDI) file:
niversity_Program/Computer_Systems/DE1-SoC/DE1-SoC_Computer/verilog/DE1_SoC_Computer.jdi Browse...

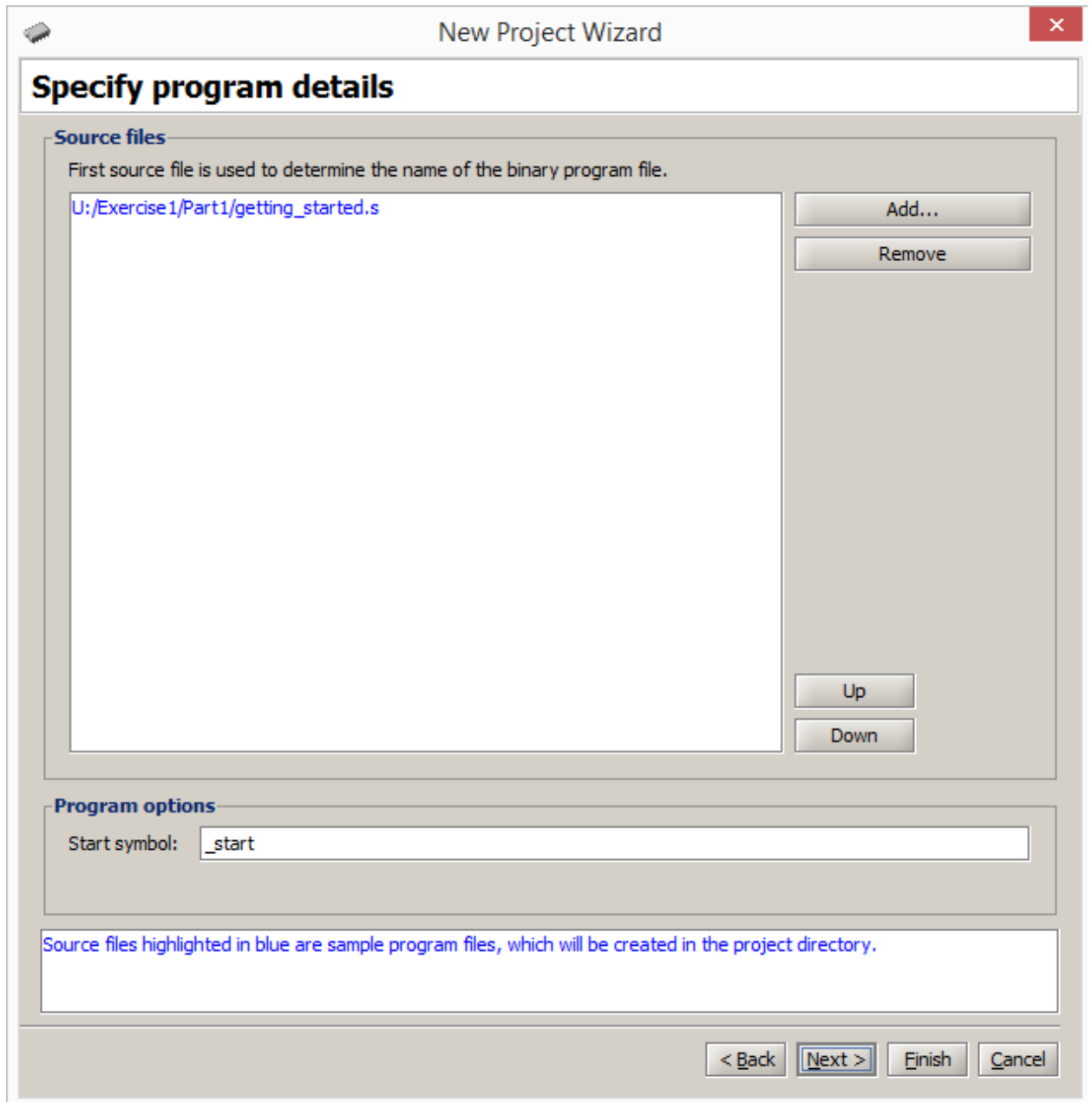
The JDI file is required for multiprocessor systems designed in Qsys. It stores the JTAG Device IDs. These IDs are needed for communication between the Monitor Program and the system's multiple processors and/or JTAG UARTs .

< Back Next > Finish Cancel

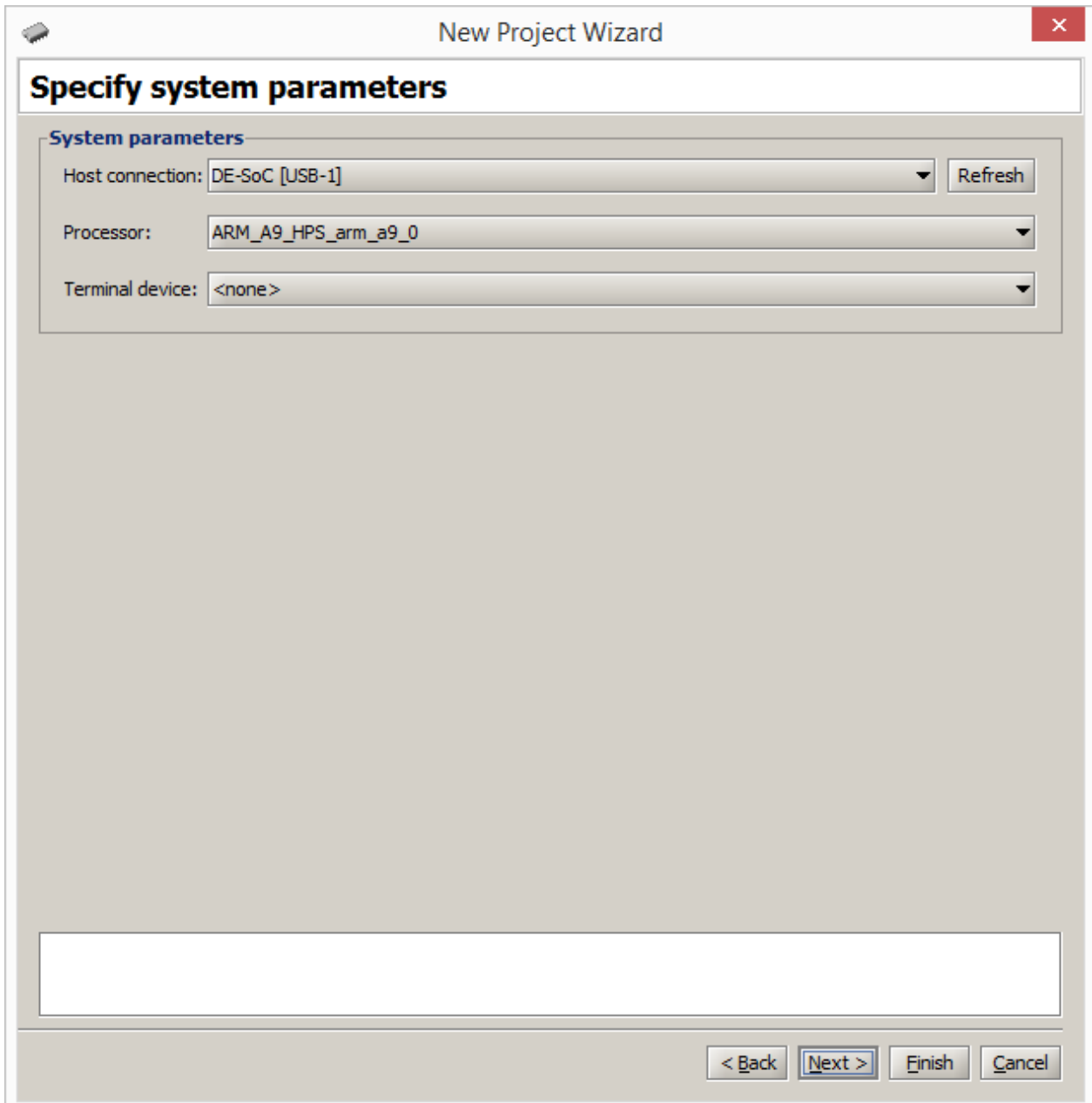
6. Выбираем Assembly language. В окне вставок выбираем Getting Started.
Next.



7. Следующее окно указывает источник кода getting started. Next.

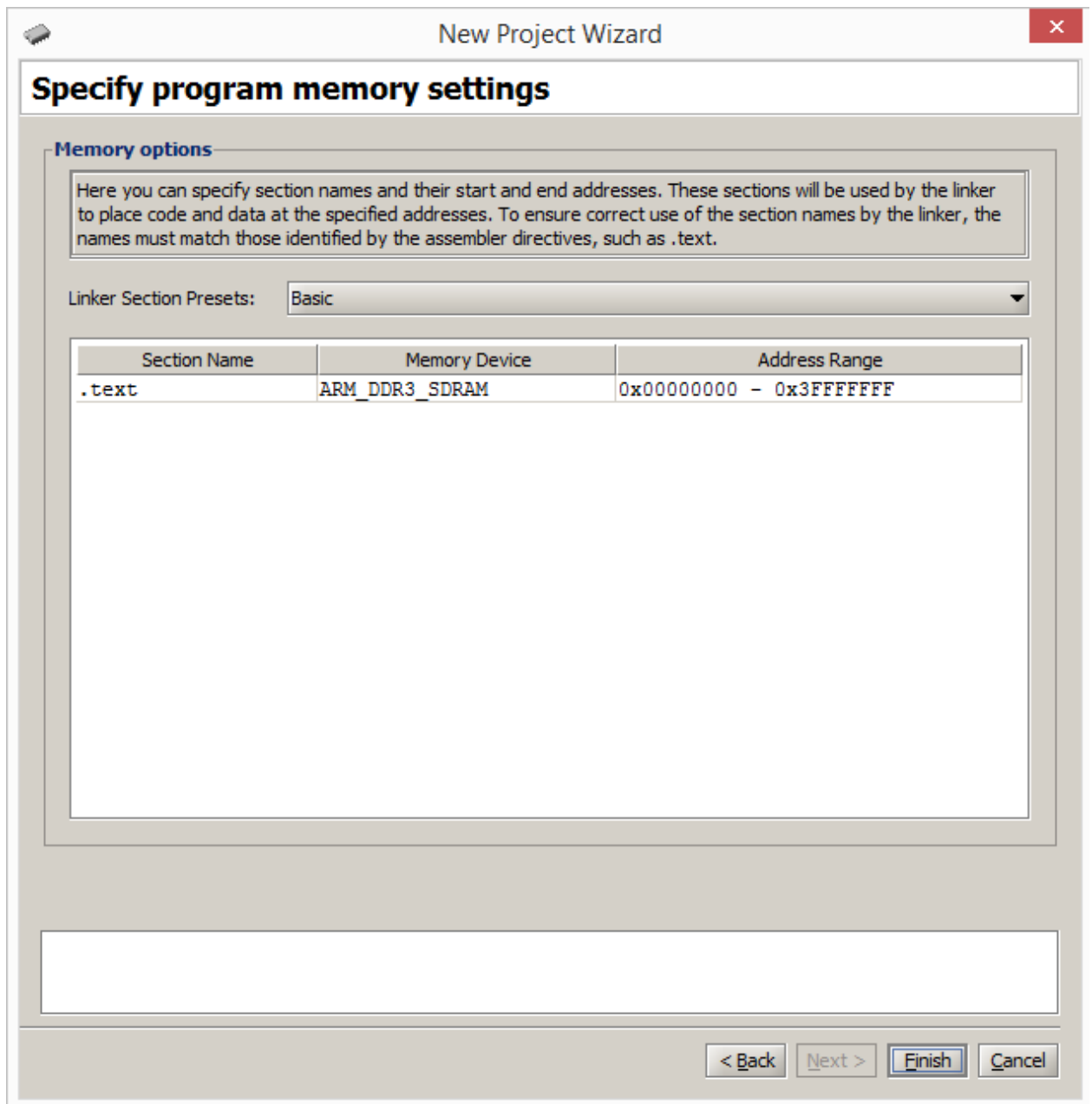


8. В окне параметров выбираем соединение DE1-SoC. Процессор ARM_A9_HPS_a9_0. Next.

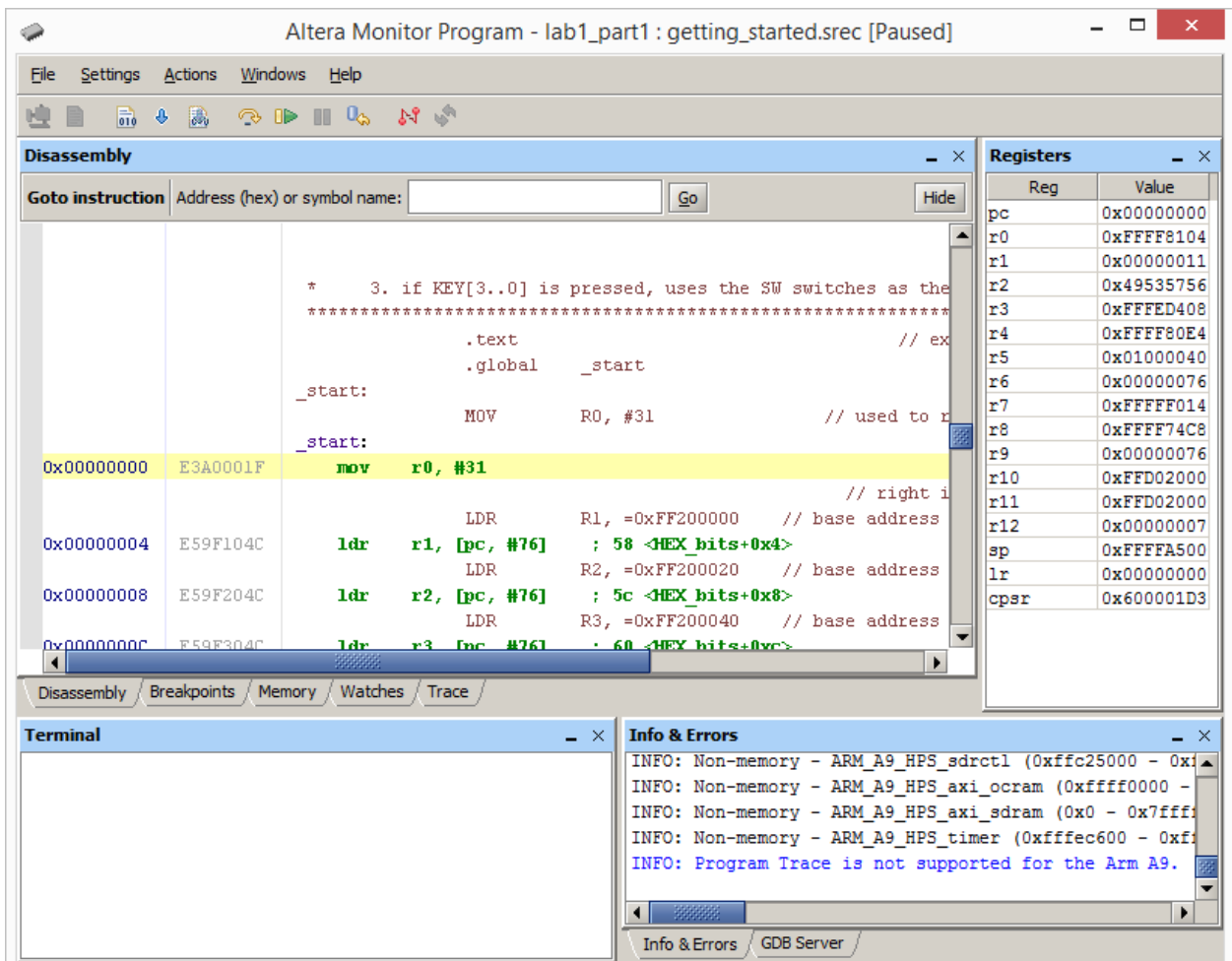


The image shows a screenshot of the 'New Project Wizard' dialog box, specifically the 'Specify system parameters' step. The window title is 'New Project Wizard' and it has a close button (X) in the top right corner. The main heading is 'Specify system parameters'. Below this, there is a section titled 'System parameters' containing three dropdown menus: 'Host connection' (set to 'DE-SoC [USB-1]'), 'Processor' (set to 'ARM_A9_HPS_arm_a9_0'), and 'Terminal device' (set to '<none>'). A 'Refresh' button is located to the right of the 'Host connection' dropdown. At the bottom of the dialog, there are four buttons: '< Back', 'Next >', 'Finish', and 'Cancel'. The 'Next >' button is highlighted with a blue border.

9. В следующем окне отображается устройство и диапазон адресов. Внизу выбираем Finish.



10. Далее будет предложена загрузка кодов в макет. Осуществить загрузку, дав положительный ответ на запрос “Download System”.



11. В появившемся окне выбрать Actions> Compile&Load. Выполнять программу пошагово, нажимая на иконку с желтой дугообразной стрелкой.

На тумблерах (T3 - T0) выставить число 11(1011).

Перед командой 9 нажать любую кнопку.

Цикл SUB_LOOP выполнить один раз, затем запустить программу на выполнение, используя иконку справа, с зеленой стрелкой.

12. После каждого шага просматривать содержимое регистров (правое окно), заполнить таблицу. **Измененные состояния регистров отображаются красным цветом!**

В отчете привести текст программы и таблицу.

Таблица

| Номер выполненной команды | Состояние программного счетчика | Измененные состояния регистров |
|---------------------------|---------------------------------|--------------------------------|
| 1. | pc = 0x00000004 | R0 = 0x..... |
| 2. | | |
| 3. | | |
| | | |
| | | |
| | | |
| 20. | | |

Текст программы:

```

1. mov r0,#31;           31 позиция для сдвигов
2. ldr r1,[pc,#76];
3. ldr r2,[pc,#76];
4. ldr r3,[pc,#76];
5. ldr r4,[pc,#76];
6. ldr r5,[pc,#56];
DO_DISPLAY:
7. ldr r6,[r3];         загрузка в r6 числа с тумблеров
8. str r6,[r1];         выдача содержимого r6 на светодиоды
9. ldr r7,[r4];         загрузка в r7 данных с кнопок
10. cmp r7,#0;          сравнение с «0»
11. beq 3c <NO BUTTON>; переход, если «0»
12. mov r5,r6;          содержимое r6 копируется в r5
WAIT:                   цикл ожидания «0» на кнопках
13.ldr r7,[r4];
14.cmp r7,#0;
15. bne 30 <WAIT>;
NO BUTTON:
16. str r5,[r2];        выдача содержимого r5 на сегменты
17. ror r5,r5,r0;       сдвиг вправо на количество позиций, указанных в r0
18. ldr r6,[pc,#28];
SUB_LOOP:               задержка, число циклов указано в r6

```

```
19. subs r6,r6,#1;
20. bne 48 <SUB_LOOP>;
21. b 18 <DO_DISPLAY>;
```

HEX_bits

```
00000054 0x0000 000f;
```

```
00000058 0xff20 0000;
```

```
0000005c 0xff20 0020;
```

```
00000060 0xff20 0040;
```

```
00000064 0xff20 0050;
```

```
00000068 0x02faf080;
```

Адреса светодиодов

Адреса сегментов

Адреса тумблеров

Адреса кнопок