|  |
| --- |
|  |
| МИНОБРНАУКИ РОССИИ |
| Федеральное государственное бюджетное образовательное учреждение  высшего образования  **«МИРЭА - Российский технологический университет»**  **РТУ МИРЭА** |

Институт Информационных Технологий

Кафедра Вычислительной Техники (ВТ)

**ОТЧЁТ ПО ПРАКТИЧЕСКОЙ РАБОТЕ № 9**

«Проектирование вычислительных устройств для синтеза сигналов трансцендентных функций»

по дисциплине

«Схемотехника устройств компьютерных систем»

|  |  |
| --- | --- |
| Выполнил студент группы  ИВБО-20-23 | Деревянных В.С. |
| Принял старший преподаватель кафедры ВТ | Дуксин Н.А. |
| Практическая работа выполнена | «\_\_»\_\_\_\_\_\_\_2025 г. |
| «Зачтено» | «\_\_»\_\_\_\_\_\_\_2025 г. |

Москва 2025

Аннотация

Работа включает в себя 11 рисунков, 5 листингов. Количество страниц в работе — 17.

Содержание

[Введение 4](#_Toc216357327)

[1 Постановка задачи 5](#_Toc216357328)

[2 Проектирование и реализация 6](#_Toc216357329)

[3 Текстирование 12](#_Toc216357330)

[Заключение 16](#_Toc216357331)

[Список использованных источников 17](#_Toc216357332)

Введение

Динамически реконфигурируемые системы (Run-Time Reconfiguration) отличаются от обычных тем, что позволяют быстро менять настройки. Стандартная загрузка конфигурации через последовательный поток бит занимает значительное время, тогда как в динамически реконфигурируемых решениях заранее сохранён набор конфигураций, которые оперативно переключаются под задачи алгоритма. Количество доступных настроек ограничено лишь объёмом памяти. Такие устройства уже применяются на практике и показывают ожидаемый результат [1].

Частичная реконфигурация — это изменение части аппаратной схемы без остановки остальной логики. Критически важные блоки продолжают работу, пока контроллер загружает новую конфигурацию в реконфигурируемую область.

1. Постановка задачи

Изучить механизм технической реконфигурации в Xilinx Vivado.

1. Проектирование и реализация

Сначала создадим логику, подлежащую замене во время работы устройства. Реализованы три варианта: конкатенация однобитных регистров, их инверсия и сложение. Все модули имеют одинаковое имя. Версии на Verilog приведены в Листингах 2.1–2.3.

Листинг 2.1 — Конкатенация регистров

|  |
| --- |
| `timescale 1ns / 1ps  module func\_block(      input a, b, clk,      output reg [1:0] c  );  reg a\_, b\_;  always@(posedge clk)  begin      a\_ <= a;      b\_ <= b;      c <= {~a\_, b\_};  end  endmodule |

Листинг 2.2 — Инверсия конкатенации регистров

|  |
| --- |
| `timescale 1ns / 1ps  module func\_block(      input a, b, clk,      output reg [1:0] c  );  reg a\_, b\_;  always@(posedge clk)  begin      a\_ <= a;      b\_ <= b;      c <= {a\_, ~b\_};  end  endmodule |

Листинг 2.3 — Сложение регистров

|  |
| --- |
| `timescale 1ns / 1ps  module func\_block(      input a, b, clk,      output reg [1:0] c  );  reg a\_, b\_;  always@(posedge clk)  begin      a\_ <= a;      b\_ <= b;      c <= b\_ + a\_;  end  endmodule |

Создадим модуль верхнего уровня, в котором размещён инстанс функционального блока и логика счётчика (Листинг 2.4).

Листинг 2.4 — Модуль верхнего уровня

|  |
| --- |
| `timescale 1ns / 1ps  module top(      input clk, a, b,      output reg [1:0] c,      output [3:0] cnt\_out  );  reg [24:0] cnt;  assign cnt\_out = cnt[24 -: 4];  reg a\_static, b\_static;  wire [1:0] c\_static;  func\_block fb(      .a(a\_static),      .b(b\_static),      .c(c\_static),      .clk(clk)  );  always@(posedge clk)  begin      a\_static <= a;      b\_static <= b;      c <= c\_static;  end  always@(posedge clk)  begin      cnt <= cnt + 1;  end  endmodule |

После синтеза схемы задаётся PBLOCK — область устройства, которой назначается инстанцированный модуль. Он объединяет нужные ресурсы для логики и фиксирует её размещение [2]. Pblock для целевого модуля показан на рисунке 2.1.

Изображение выглядит как текст, снимок экрана, программное обеспечение, Мультимедийное программное обеспечение

Содержимое, созданное искусственным интеллектом, может быть неверным.

Рисунок 2.1 — Присвоенный Pblock

После добавления PBLOCK экспортируем проектные ограничения (рисунок 2.2), подключаем их в проект и дополняем необходимыми параметрами [3]. Файл ограничений приведён в Листинге 2.5.

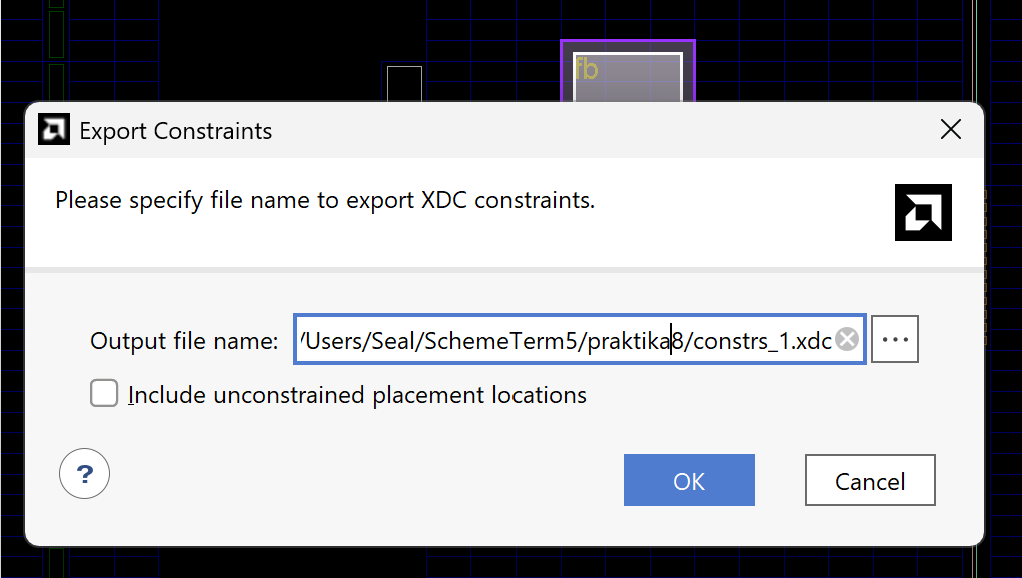


Рисунок 2.2 — Экспорт файла проектных ограничений

Листинг 2.4 — Файл проектных ограничений

|  |
| --- |
| create\_pblock pblock\_1  add\_cells\_to\_pblock [get\_pblocks pblock\_1] [get\_cells -quiet [list fb]]  resize\_pblock [get\_pblocks pblock\_1] -add {SLICE\_X0Y65:SLICE\_X5Y70}  ## Clock signal  set\_property -dict {PACKAGE\_PIN E3 IOSTANDARD LVCMOS33} [get\_ports clk]  create\_clock -period 10.000 -name sys\_clk\_pin -waveform {0.000 5.000} -add [get\_ports clk]  ##Switches  set\_property -dict {PACKAGE\_PIN J15 IOSTANDARD LVCMOS33} [get\_ports a]  set\_property -dict {PACKAGE\_PIN L16 IOSTANDARD LVCMOS33} [get\_ports b]  ## LEDs  set\_property -dict {PACKAGE\_PIN H17 IOSTANDARD LVCMOS33} [get\_ports {c[0]}]  set\_property -dict {PACKAGE\_PIN K15 IOSTANDARD LVCMOS33} [get\_ports {c[1]}]  set\_property -dict {PACKAGE\_PIN V15 IOSTANDARD LVCMOS33} [get\_ports {cnt\_out[0]}]  set\_property -dict {PACKAGE\_PIN V14 IOSTANDARD LVCMOS33} [get\_ports {cnt\_out[1]}]  set\_property -dict {PACKAGE\_PIN V12 IOSTANDARD LVCMOS33} [get\_ports {cnt\_out[2]}]  set\_property -dict {PACKAGE\_PIN V11 IOSTANDARD LVCMOS33} [get\_ports {cnt\_out[3]}] |

Далее включаем механизм динамической реконфигурации (рисунок 2.3) и создаём частично реконфигурируемый модуль на основе подключённого блока (рисунок 2.4).

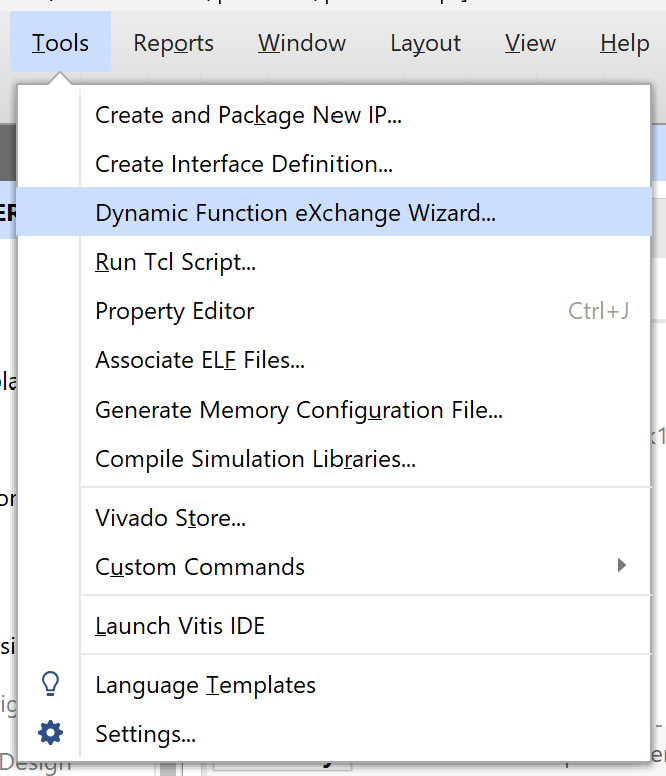


Рисунок 2.3 — Включение динамической реконфигурации



Рисунок 2.4 — Создание частичного определения

В Dynamic Function eXchange Wizard выполняется настройка реконфигурации. На рисунке 2.5 показано добавление модуля сложения; аналогично добавлены варианты конкатенации и конкатенации с инверсией.

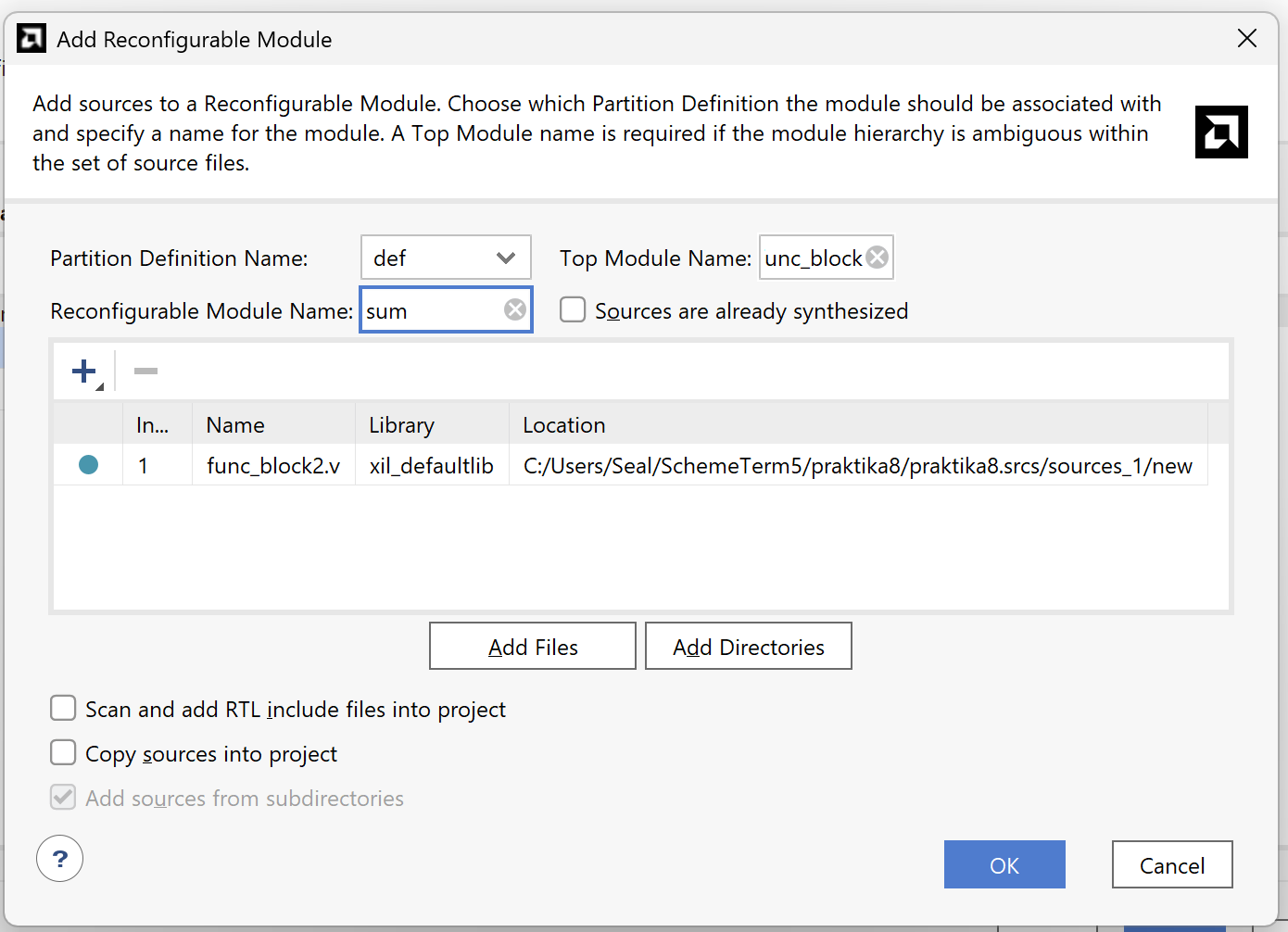


Рисунок 2.5 — Добавление модуля суммы

Конфигурации привязываются к выбранным модулям (рисунок 2.6), затем создаются Configuration Runs для предварительной имплементации (рисунок 2.7). После этого выполняется генерация bitstream (рисунок 2.8).

Изображение выглядит как текст, снимок экрана, дисплей, программное обеспечение

Содержимое, созданное искусственным интеллектом, может быть неверным.

Рисунок 2.6 — Добавление конфигураций

Изображение выглядит как текст, снимок экрана, программное обеспечение, Значок на компьютере

Содержимое, созданное искусственным интеллектом, может быть неверным.

Рисунок 2.7 — Добавление Configuration Runs

Изображение выглядит как текст, снимок экрана, программное обеспечение, линия

Содержимое, созданное искусственным интеллектом, может быть неверным.

Рисунок 2.8 — Сообщение о завершении генерации bitstream

1. Текстирование

Проверка проводилась на отладочной плате. Четыре левых светодиода отображают значение счётчика, два правых — результат работы логического модуля. Входные данные задаются двумя правыми переключателями. Результаты тестирования для всех вариантов модулей приведены на рисунках 3.1–3.3.

Изображение выглядит как электроника, Электронная техника, Компьютерный компонент, Электронный компонент

Содержимое, созданное искусственным интеллектом, может быть неверным.

Рисунок 3.1 — Модуль конкатенации

Изображение выглядит как электроника, Электронная техника, Электронный компонент, Компьютерный компонент

Содержимое, созданное искусственным интеллектом, может быть неверным.

Рисунок 3.2 — Модуль конкатенации с инверсией

Изображение выглядит как электроника, Электронная техника, Электронный компонент, Компонент схемы

Содержимое, созданное искусственным интеллектом, может быть неверным.

Рисунок 3.3 — Модуль сложения

Заключение

В ходе работы изучен процесс динамической частичной реконфигурации и выполнена практическая реализация механизма.

Список использованных источников

1. Угрюмов Е. П. Цифровая схемотехника: учеб. пособие для вузов. — 3-е изд., перераб. и доп. — СПб.: БХВ-Петербург, 2010. — 816 с.: ил. — ISBN 978-5-9775-0162-0
2. AMD Technical Inforamation Portal — URL: https://docs.amd.com/r/en-US/ug912-vivado-properties/PBLOCK (Дата обращения: 05.12.2025) — Текст: электронный
3. Github — URL: https://github.com/Digilent/digilent-xdc (Дата обращения: 05.12.2025) — Текст: электронный
4. AMD Technical Inforamation Portal — URL: https://docs.amd.com/r/en-US/ug909-vivado-partial-reconfiguration/Running-the-Dynamic-Function-eXchange-Wizard (Дата обращения: 05.12.2025) — Текст: электронный