Выполнение лабораторной работы «**lab\_scr1\_rtl**»

Таблица 1 Условие согласно варианту № 7

|  |  |  |  |
| --- | --- | --- | --- |
| №  вар | Instr | Arch #1 | Arch #2 |
| 7 | **JALR** | **RVIM -FAST\_MUL** | **RVIM +FAST\_MUL** |

**Часть I**

Для выполнения данной части был запущен соответствующий названию инструкции тест *isa/rv32ui/jalr.S*.

На осциллограмме был найден отрезок выполнения исследуемой инструкции (рис. 1).

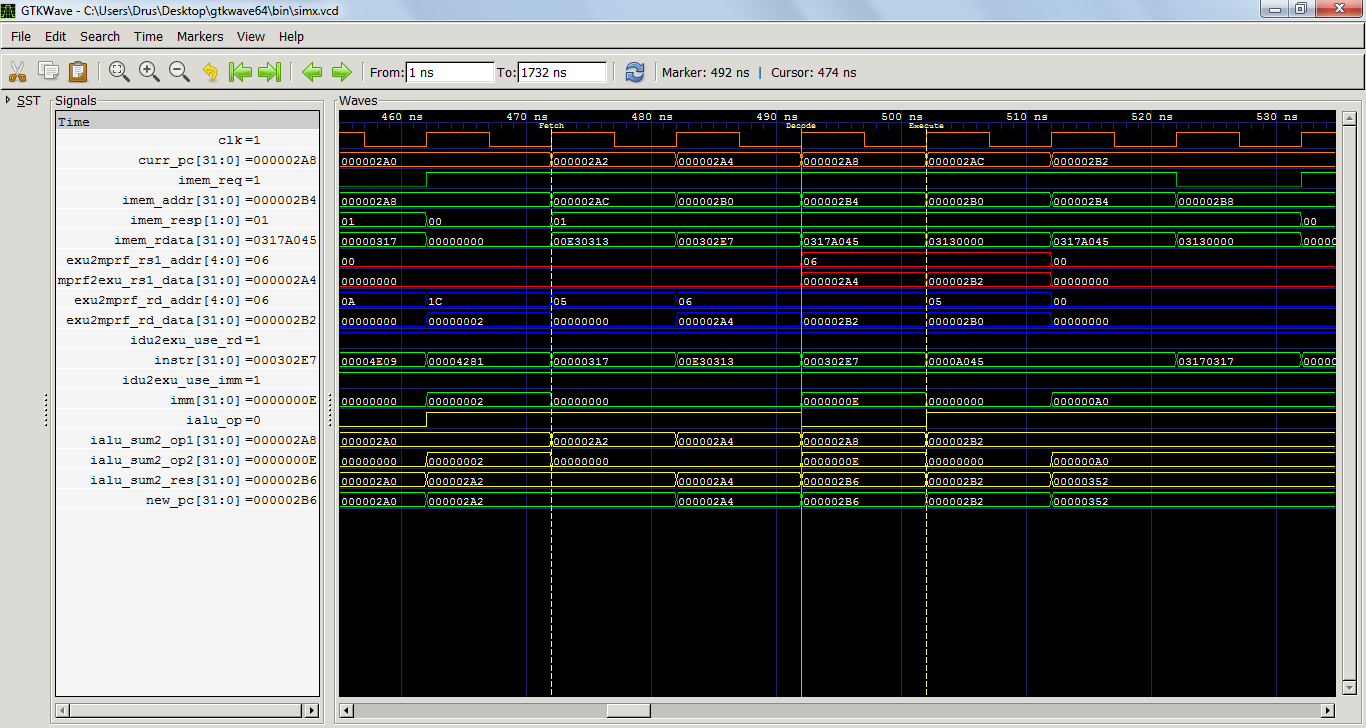


Рисунок 1 Отрезок выполнения инструкции JALR

Этому отрезку соответствует следующая часть дампа теста из файла *.results/jalr.dump*, представленная на рисунке 2, и часть из файла *jalr.s*, представленная на рисунке 3.

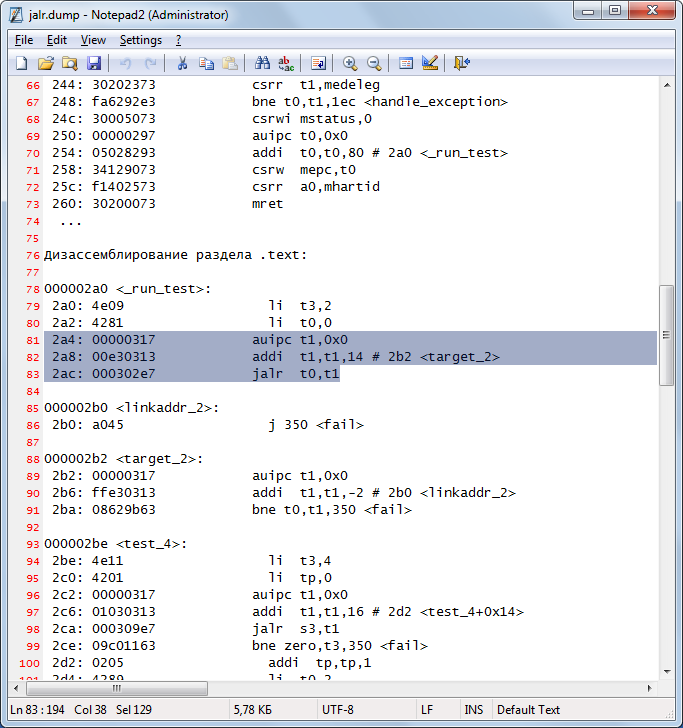


Рисунок 2 Часть дампа выполнения теста

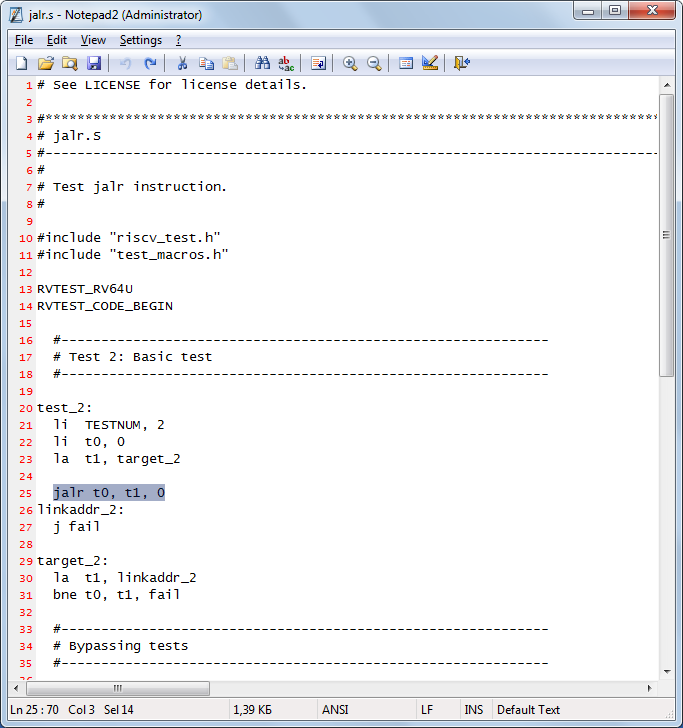


Рисунок 3 Исследуемая часть кода теста

Стадия выборки команды отмечена на осциллограмме маркером «Fetch». На этой стадии получен ответ памяти инструкций (imem\_resp = 01), значение imem\_addr соответствует адресу из дампа (рис. 2), по которому расположена исполняемая инструкция. На следующем такте в imem\_rdata располагается код инструкции. Для подтверждения правильности COP данная инструкция изучена в симуляторе Venus (рисунок 4).

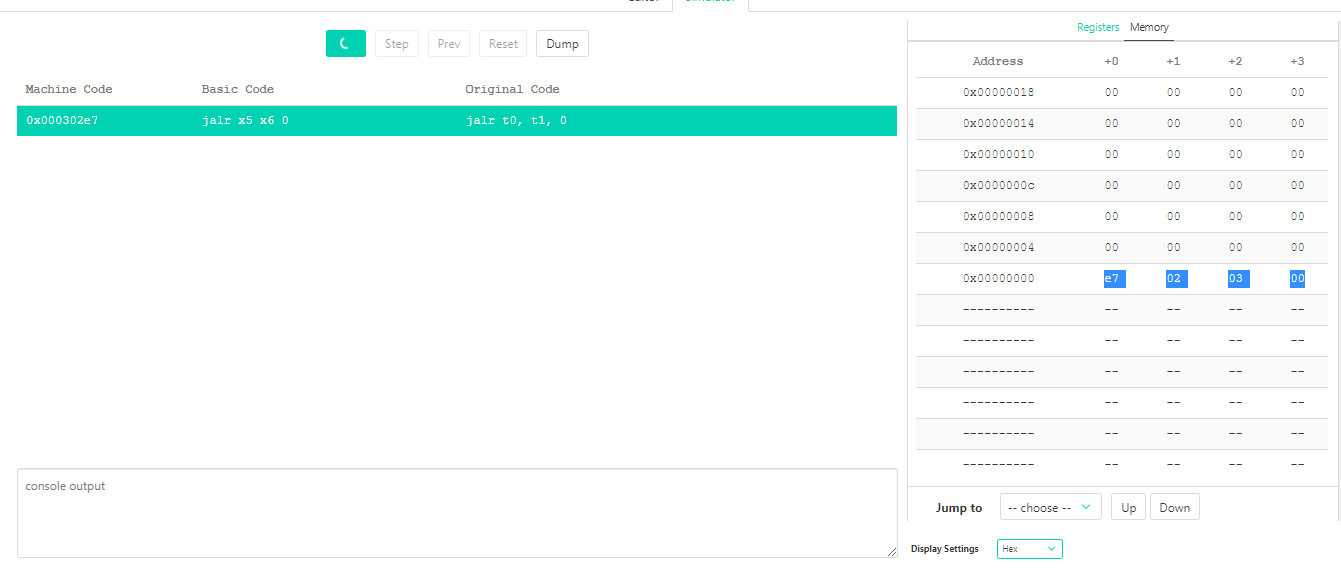


Рисунок 4 Выполнение инструкции в симуляторе Venus для проверки кода операции

Стадия декодирования инструкции отмечена, как «Decode». На осциллограмме можно наблюдать, что адрес регистра rs1 в регистровой памяти (exu2mprf\_rs1\_addr = 06) соответствует регистру x6 (он же t1). Значение адреса регистра назначения rd (exu2mprf\_rd\_addr = 05) x5(t0).

Стадия исполнения инструкции – «Execute». Ее выполнение происходит при curr\_pc = 0x2AC, что соответствует дампу. На этой стадии new\_pc = 0x2B2, что, согласно дампу, является адресом следующей исполняемой команды и метки, на которую был выполнен «jump».

**Часть II**

Для выполнения второй части лабораторной работы установим архитектуру в соответствии с вариантом в файле src/includes/scr1\_arch\_description.svh: закомментируем параметр `define SCR1\_FAST\_MUL (рисунок 5).

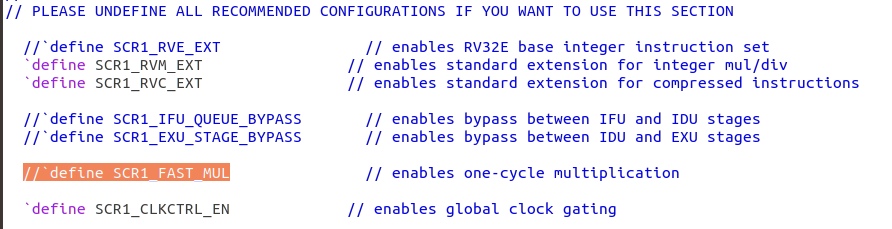


Рисунок 5 Изменение конфигурации ядра в соответствии с Arch #1

Выполним тесты Coremark и Dhrystone, результаты которых представлены на рисунках 6 и 7 соответственно.

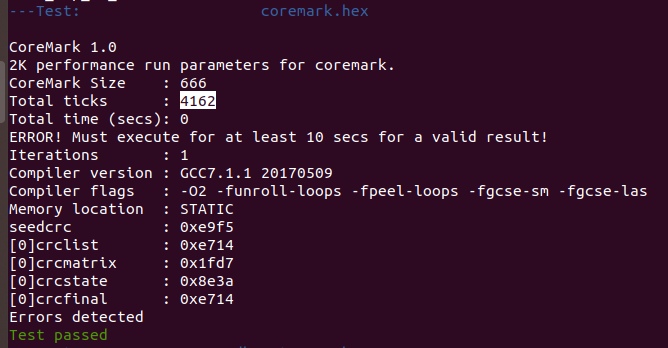


Рисунок 6 Результат выполнения теста Coremark с конфигурацией Arch #1

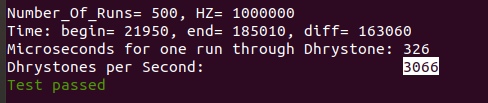


Рисунок 7 Результат выполнения теста Dhrystone с конфигурацией Arch #1

Для выполнения тестирования с конфигурацией Arch #2 необходимо раскомментировать в файле src/includes/scr1\_arch\_description.svh параметр `define SCR1\_FAST\_MUL (рисунок 8).

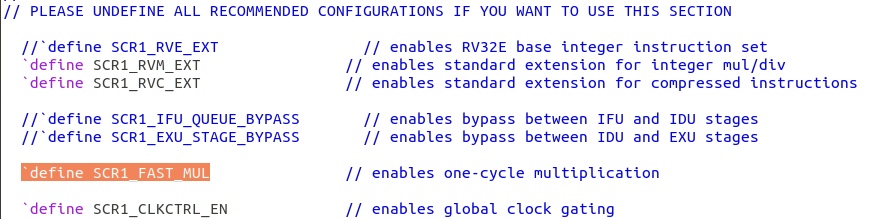


Рисунок 8

Результаты выполнения тестов Coremark и Dhrystone представлены на рисунках 9 и 10 соответственно.

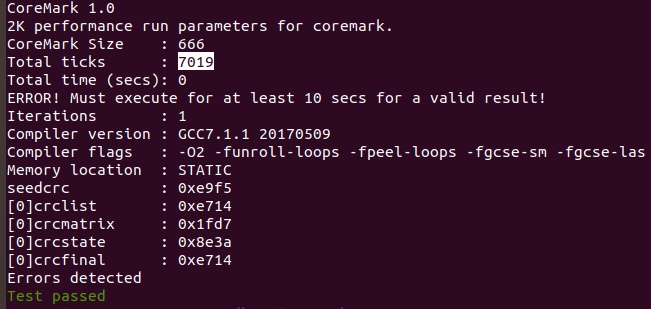


Рисунок 9 Результат выполнения теста Coremark с конфигурацией Arch #2

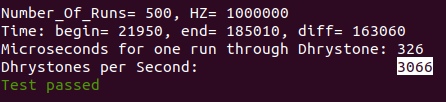


Рисунок 10 Результат выполнения теста Dhrystone с конфигурацией Arch #2

Результаты сравнения тестов при разных конфигурациях приведены в таблице 2.

Таблица 2

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Архитектура | Производительность | | Объем занимаемой памяти, КБайт | |
| Coremark, total ticks | Dhrystone, dhrystones per second | Coremark | Dhrystone |
| Arch #1 | 4162 | 3066 | 89,9 | 45,6 |
| Arch #2 | 7019 | 3066 | 89,9 | 45,6 |

**Часть III**

**III.0**

Для ознакомления с алгоритмом работы в среде Vivado был запущен проект с конфигурацией “по умолчанию” – IMC\_MAX.

На разборе лабораторной работы была выявлена максимальная тактовая частота – 33.(3) MHz (Рисунок 11). Она изменяется следующим образом:

1. Open Block Design.
2. sys\_pll.pd.
3. Даблклик по модулю clk\_wiz\_0 (откроется окно рис. 11).
4. Во вкладке Output Clocks следует изменить частоту объекта clk\_out1 в поле Requested.
5. Нажать кнопку “ОК”.

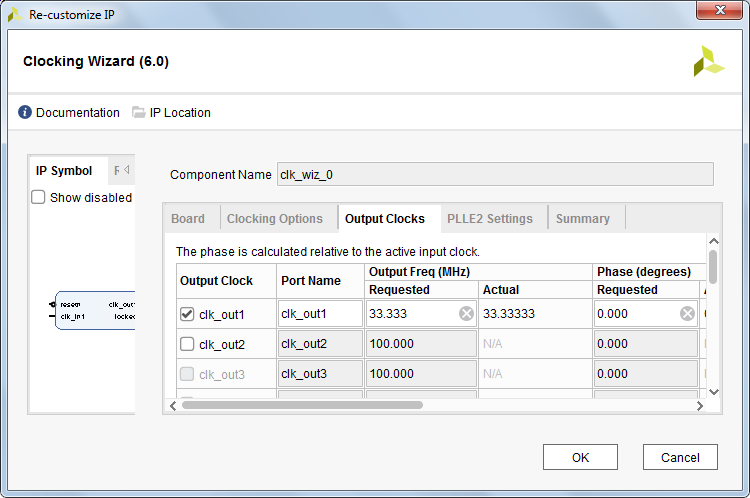


Рисунок 11 Изменение тактовой частоты

После изменения тактовой частоты необходимо задать соответствующий период в ограничениях проекта: в директории constrs\_1 в файле arty\_scr1\_synth.xdc изменить период сигнала SYS\_CLK\_VIRT. Период, соответствующий заданной ранее тактовой частоте равен 30 ns (рисунок 12). Также в параметре “-waveform” необходимо задать диапазон от 0 до константы “”.

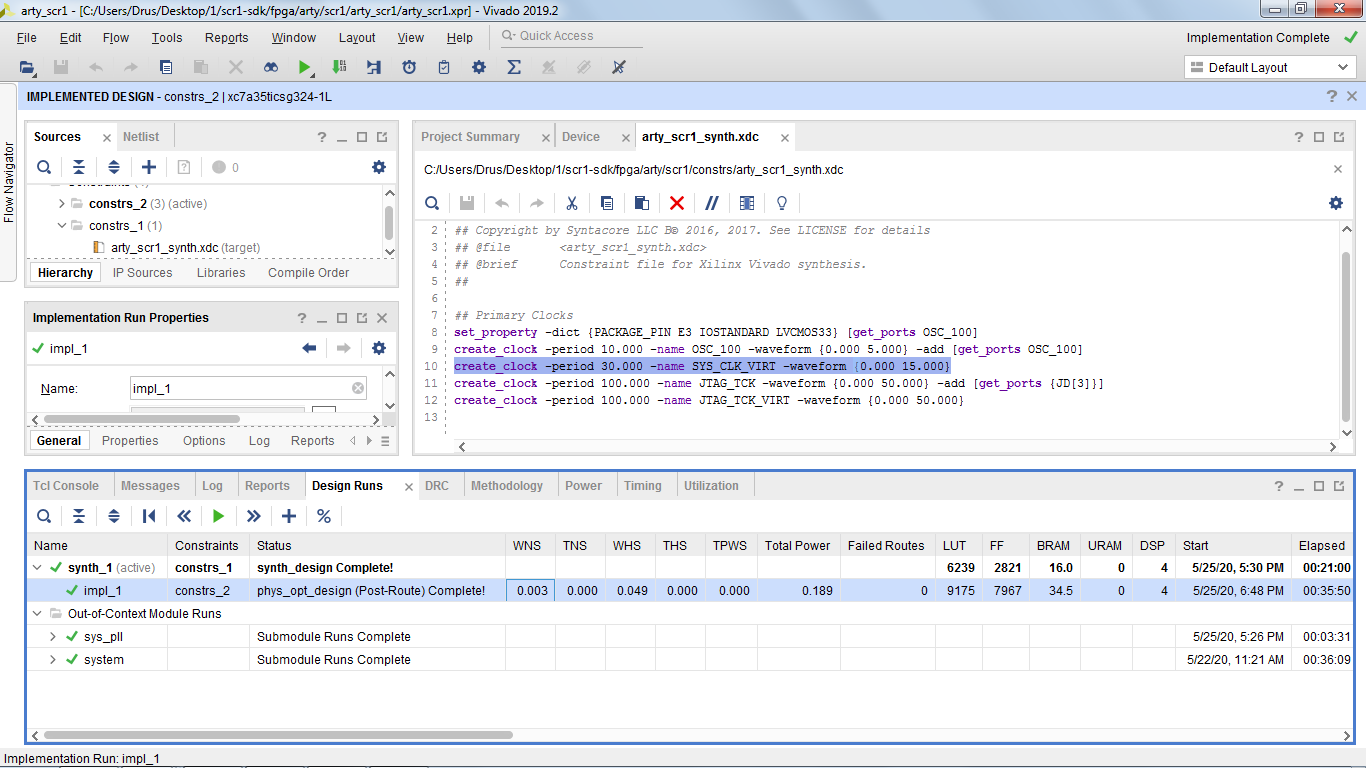


Рисунок 12 Период, соответствующий тактовой частоте 33,(3) MHz и основные показатели после запуска операции Generate Bitstream

Чтобы убедиться в том, что заданная частота оптимальна, необходимо запустить полный flow и проверить параметр WNS. Положительный WNS, он же SETUP SLACK, говорит нам о том, что отсутствуют TIMING VIOLATIONS, потому что SETUP SLACK – это не что иное, как разница между “требуемым временем на исполнение комбинаторной логики” () и “временем прибытия управляющего сигнала” (). Соответственно, если WNS (SETUP SLACK) является положительным, то это означает, что самый длинный путь проходит – отсутствуют TIMING VIOLATIONS (). Если , значит, есть TIMING VIOLATIONS. Исходя из этого частота, соответствующая периоду, при котором отсутствуют TIMING VIOLATIONS, – такая частота, при которой параметр WNS больше 0 (). Максимальной же считается частота, при которой данный параметр почти равен 0, но больше него. На рисунке 12 можно наблюдать, что WNS является положительным и почти равным 0.

На рисунке 13 представлен отчет по использованной площади платы. Исследуемый модуль SCR1 занимает наибольшую площадь.

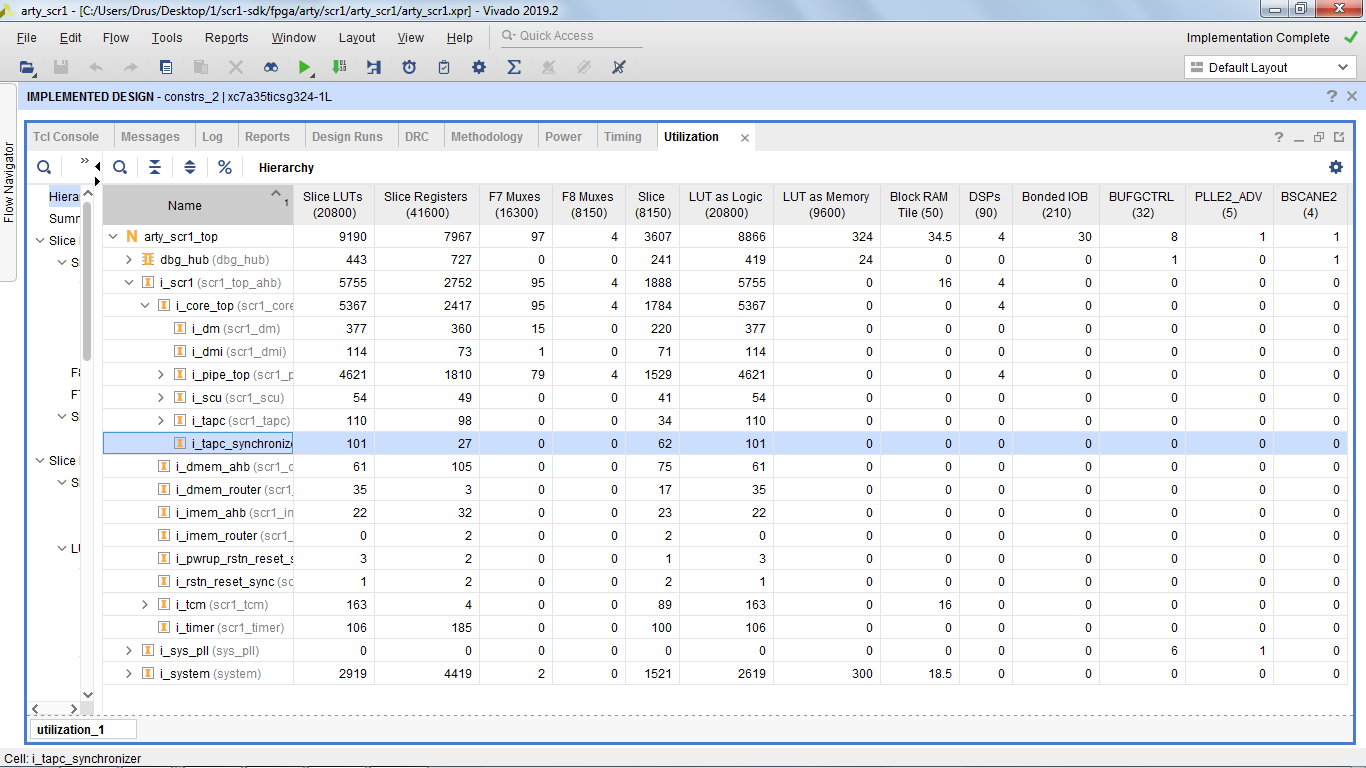


Рисунок 13 Отчет о занимаемой площади

**III.1**

Для выполнения задания по варианту необходимо изменить конфигурацию проекта по варианту: RVIM -FAST\_MUL. Соответственно, как уже было описано в части II, необходимо снести изменения в файле src/includes/scr1\_arch\_description.svh. Это можно сделать непосредственно в среде разработки Vivado (рисунок 14) закомментировав параметр IMC\_MAX и все включения кроме SCR1\_RVM\_EXT, которое предоставляет возможность умножения и деления.

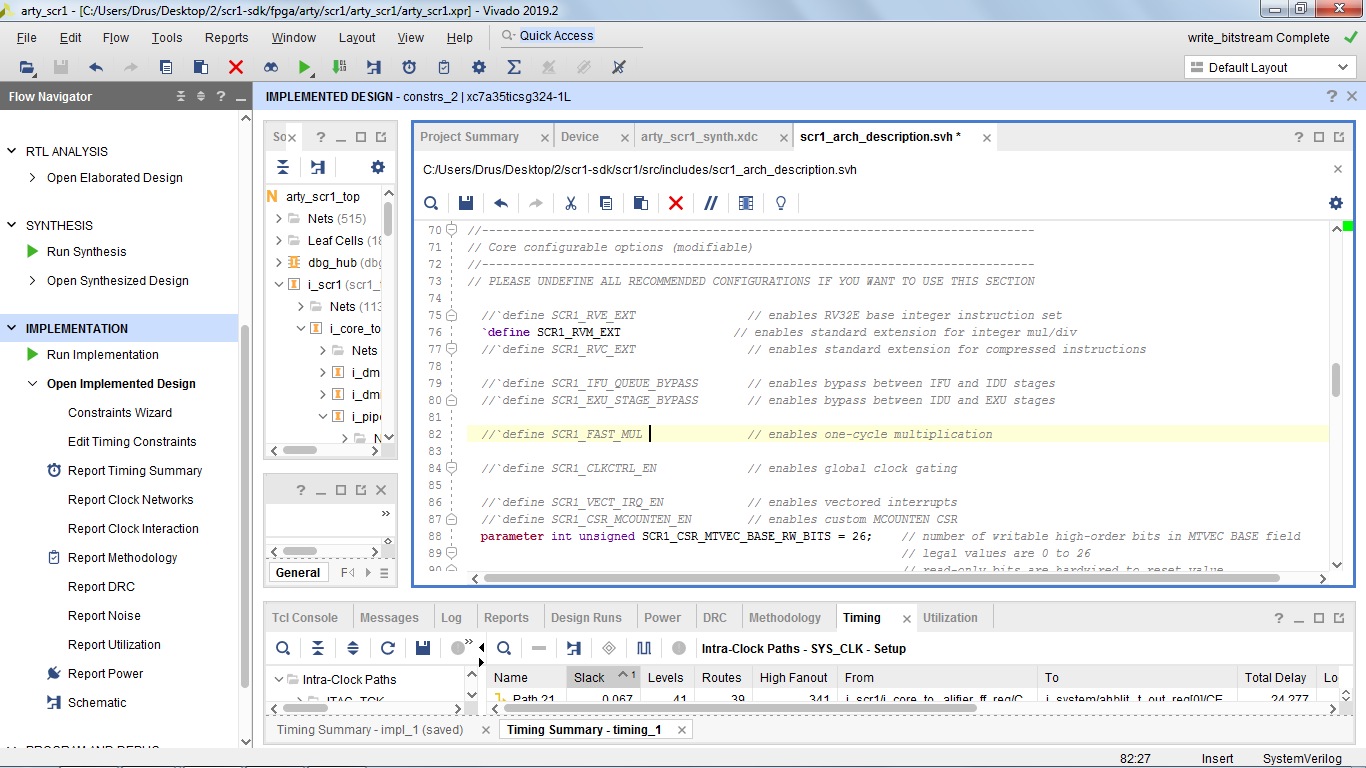


Рисунок 14 Изменения конфигурации

При выполнении операции Generate Bitstream для данной конфигурации были получены результаты, представленные на рисунках 15-16.

На рисунке 15 представлена занимаемая площадь. Можно наблюдать, что она заметно уменьшилась, так как была “уменьшена” сама конфигурация.

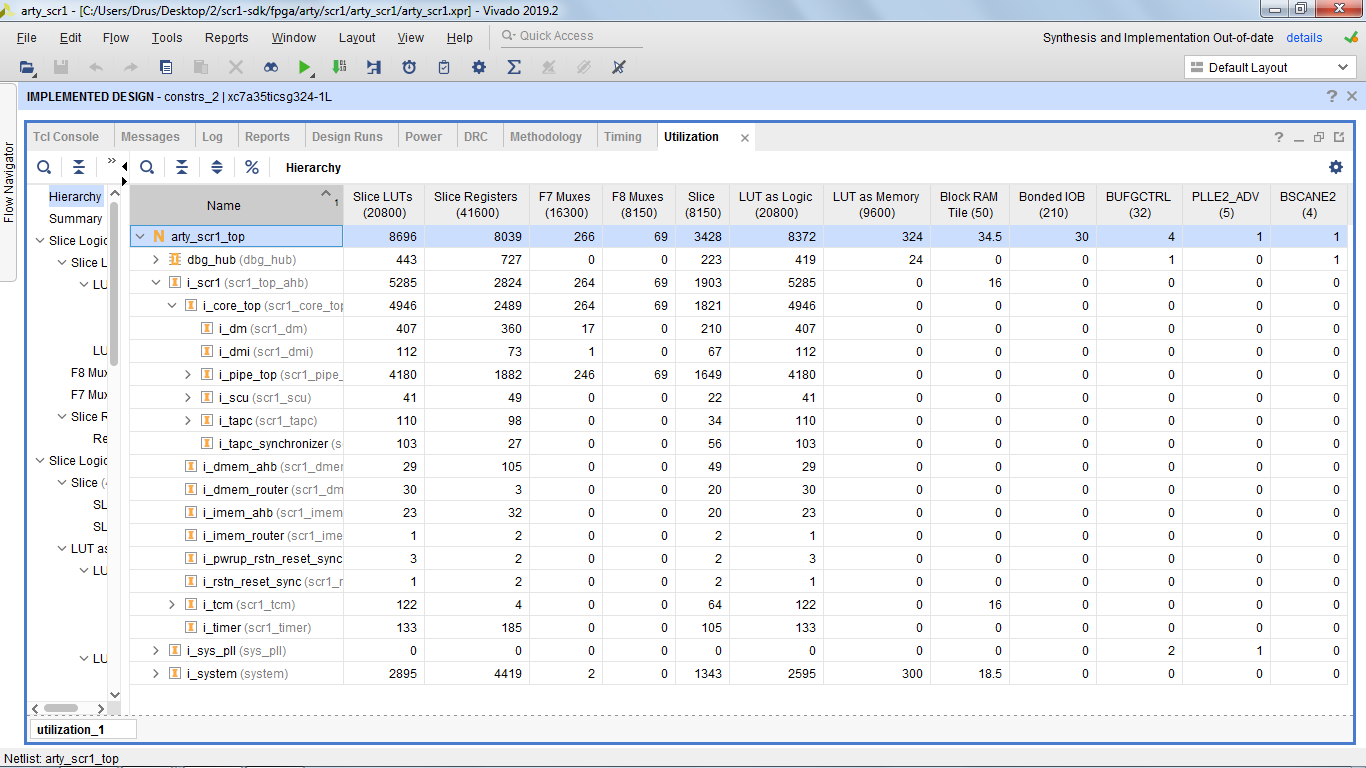


Рисунок 15 Занимаемая площадь при конфигурации -FAST\_MUL и при дефолтной частоте

На рисунке 16 можно наблюдать, что параметр WNS (SETUP SLACK) хоть и больше 0, но и не меньше 1, следовательно, при тактовой частоте по умолчанию (25 MHz) хоть и отсутствуют TIMING VIOLATIONS, но она не является максимальной.

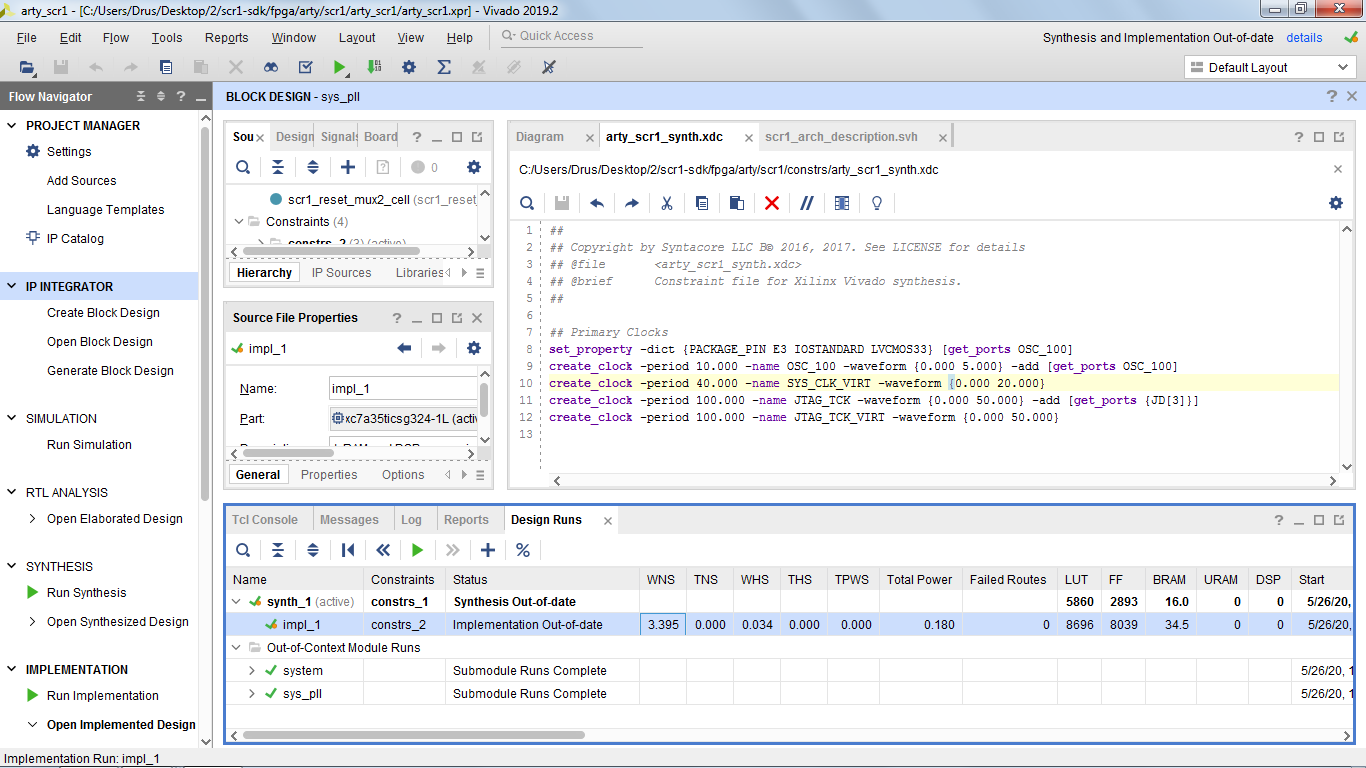


Рисунок 16 Показатель WNS и соответствующее значение периода

Для достижения максимальной частоты необходимо произвести действия, описанные в подразделе III.0. Последовательные изменения, предпринятые в ходе выполнения, представлены на рисунках 17-19.

На рисунке 17 продемонстрировано изменение тактовой частоты. 25 -> 40 MHz (заданному периоду 25 ns соответствует частота 40 MHz : ).

На рисунке 18 продемонстрировано изменение ограничений в файле arty\_scr1\_synth.xdc. Параметром –period задаем период, соответствующий заданной тактовой частоте (40 MHz > 25 ns), параметром –waveform задаем диапазон от 0 до

На рисунке 19 продемонстрировано изменение параметров запуска процесса Generate Bitstream. Для этого необходимо во вкладке Design Runs нажать ПКМ на impl\_1 и выбрать открывшемся контекстном меню “Change Run Settings…”. Задать стратегию “ExtraTimingOpt” и внести изменения в разделе Post-Route Phys Opt Design (рисунок 19).

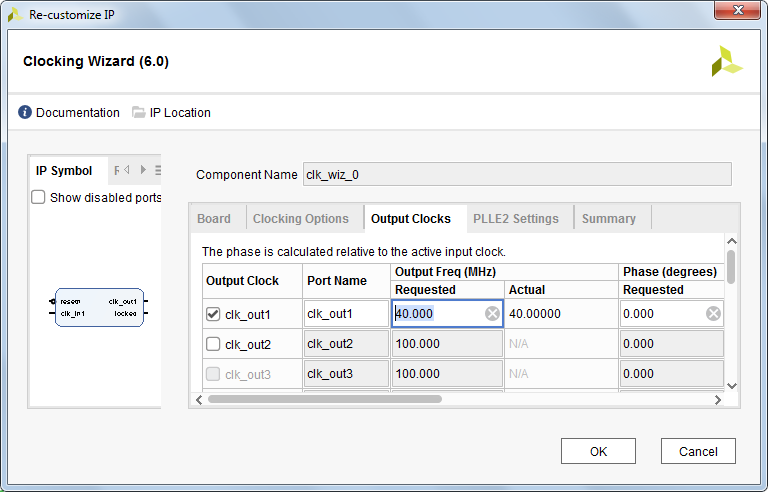


Рисунок 17 Изменение тактовой частоты clk\_wiz\_0

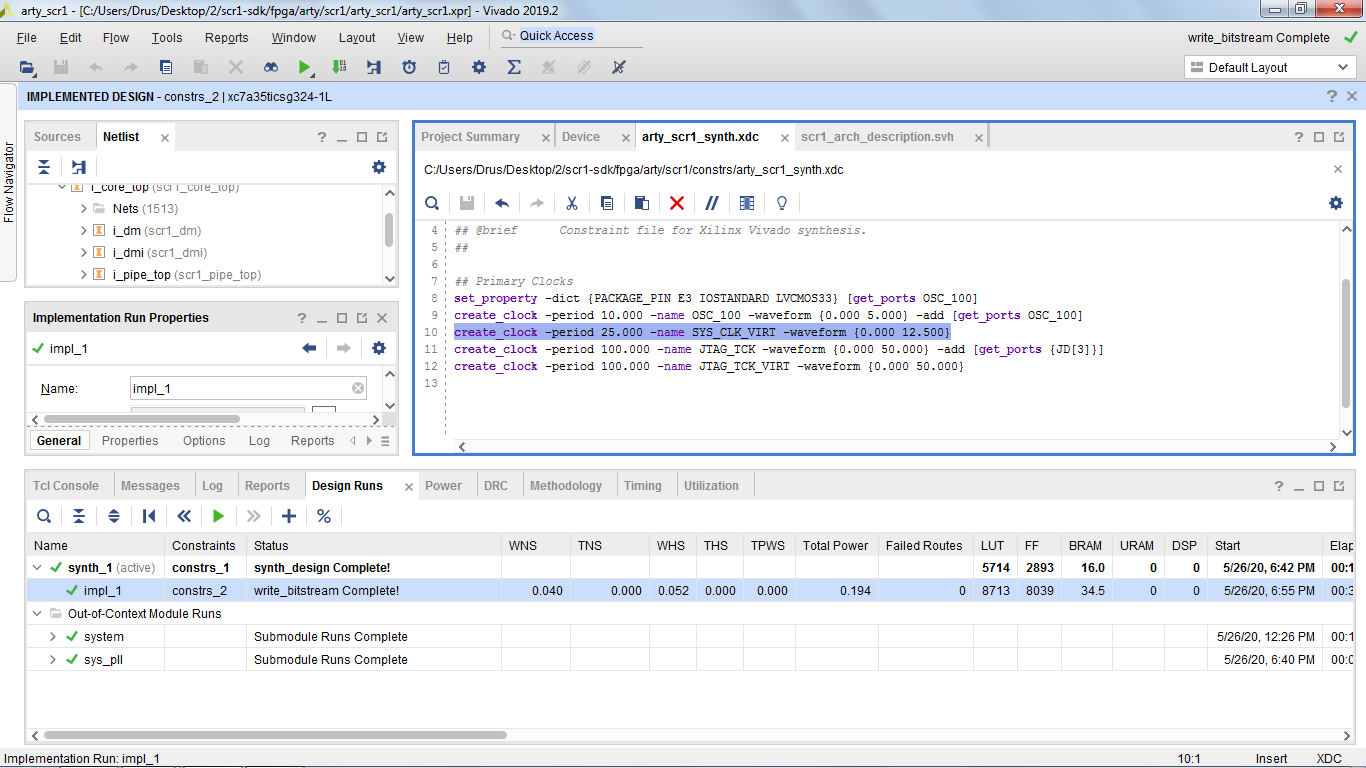


Рисунок 18 Изменение периода SYS\_CLK\_VIRT

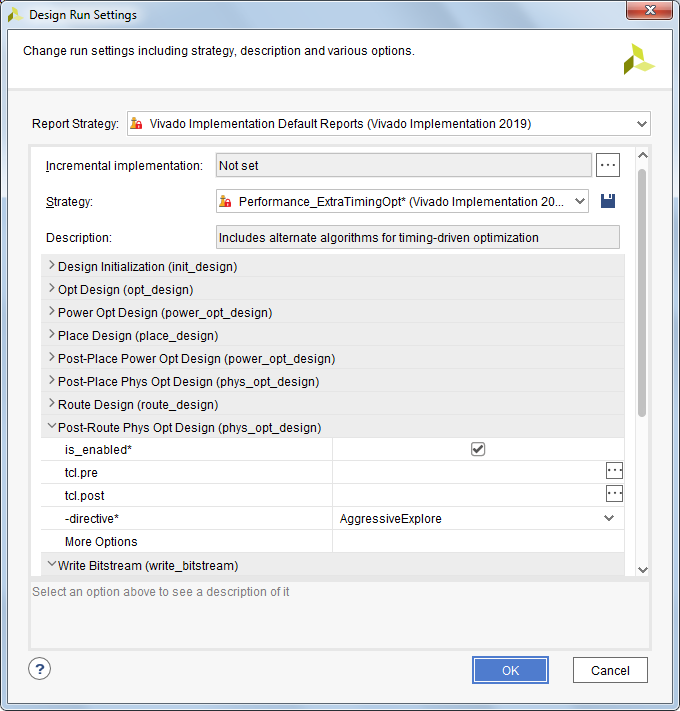


Рисунок 19 Изменение стратегии реализации

На рисунке 20 продемонстрирован показатель WNS, который соответствует максимальной тактовой частоте 40 MHz после выполнения операции Generate Bitstream.

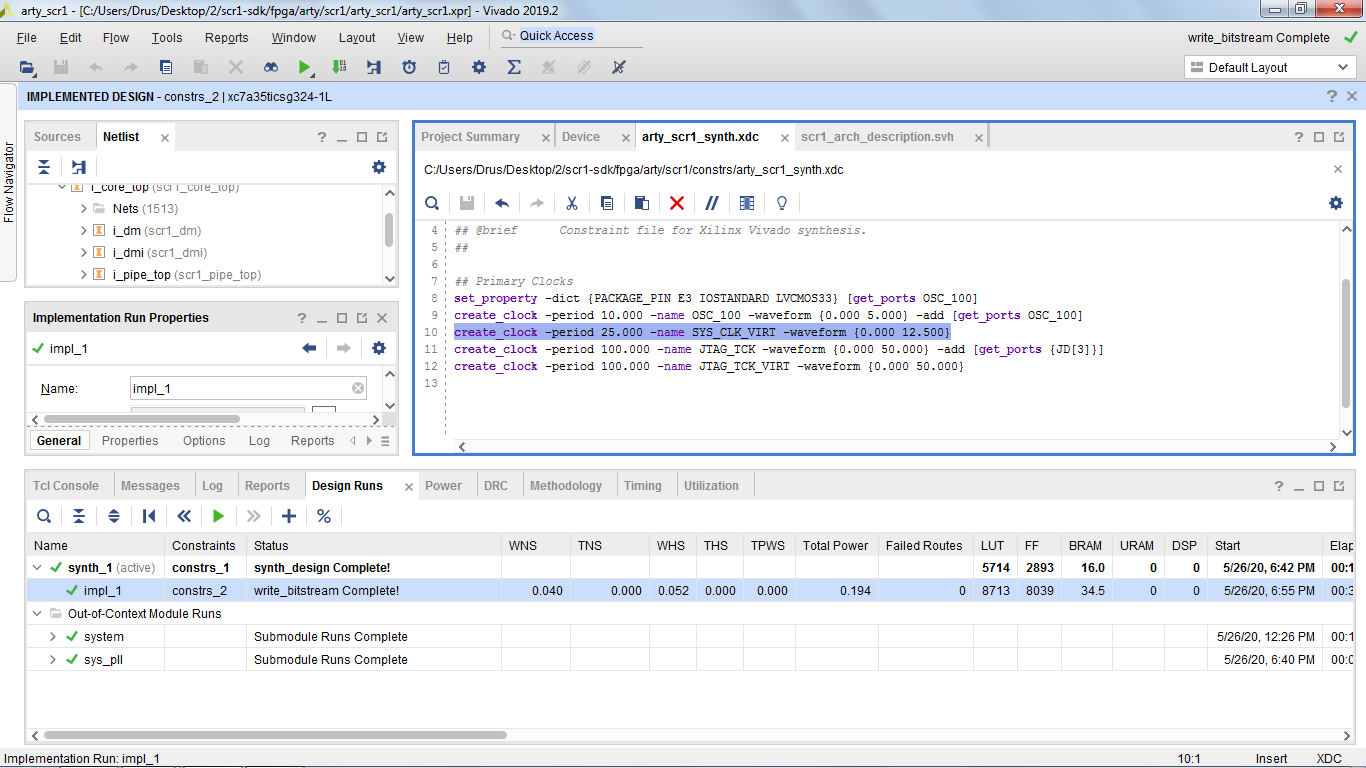


Рисунок 20 Показатель WNS после выполнения операции Generate Bitstream при тактовой частоте 40 MHz и новой стратегии реализации

На рисунке 21 продемонстрирована занимаемая площадь при внесенных раннее изменениях.

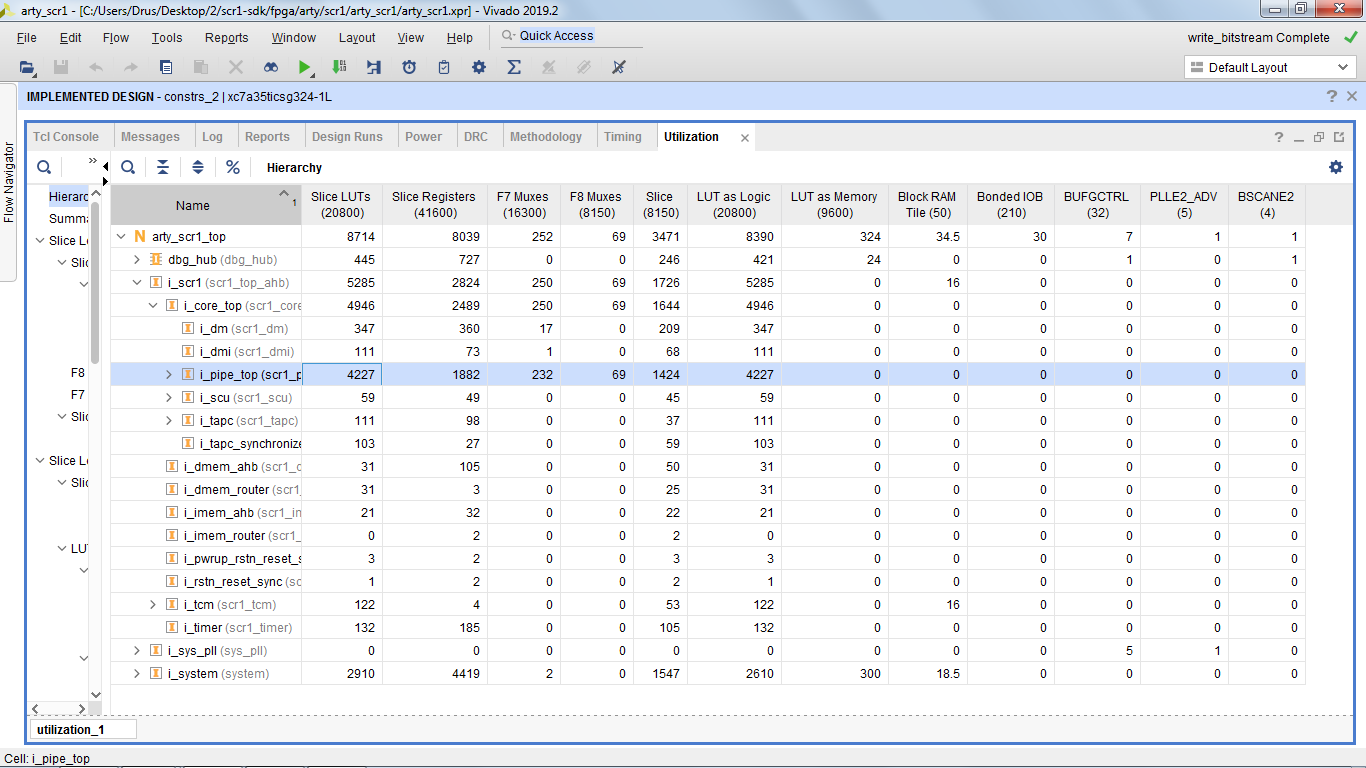


Рисунок 21 Занимаемая площадь при тактовой частоте 40 MHz

**III.2**

Для сравнения со второй конфигурацией по варианту RVIM +FAST\_MUL, аналогично подразделу III.1, необходимо изменить файл src/includes/scr1\_arch\_description.svh. Для этого в среде разработки Vivado (рисунок 22) раскомментируем параметр SCR1\_FAST\_MUL.

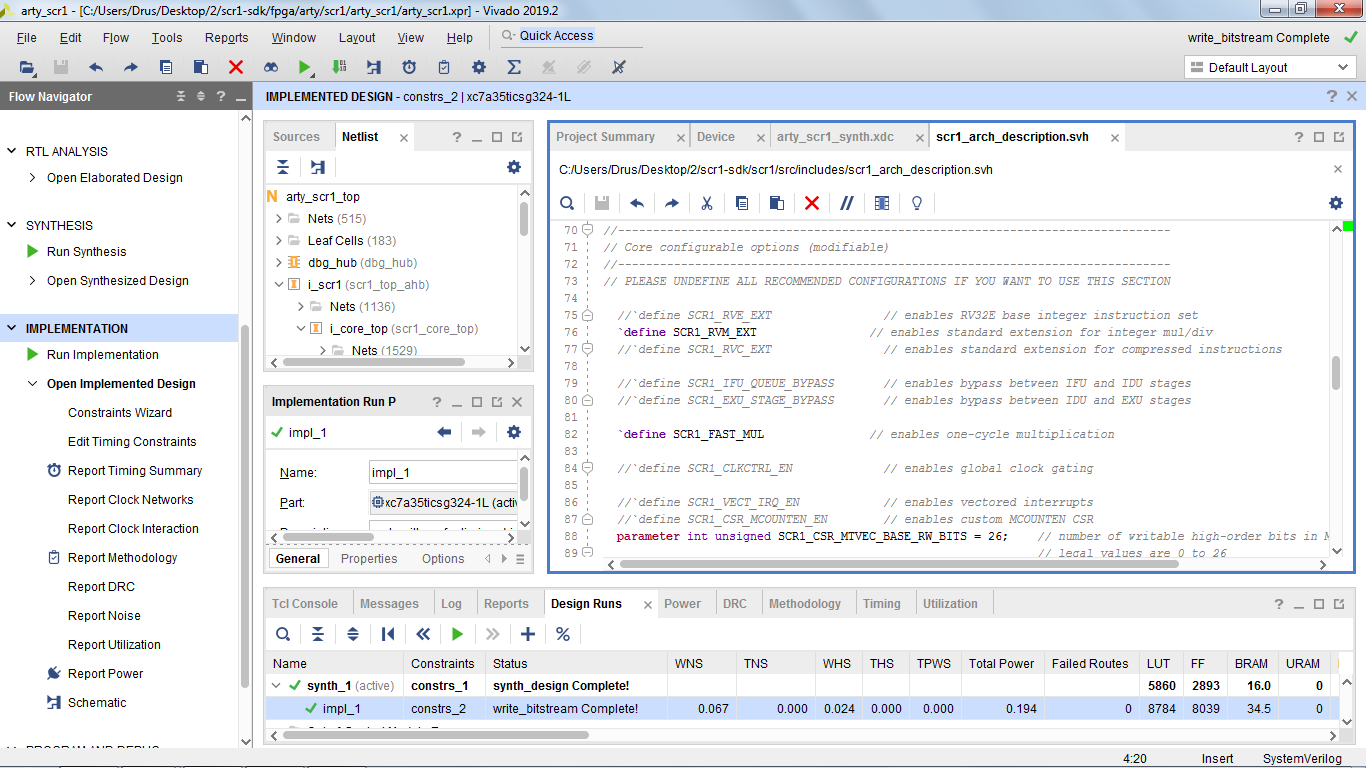


Рисунок 22 Изменение конфигурации

На рисунке 23 продемонстрирован результат запуска операции Generate Bitstream с параметрами, измененными в разделе III.1 (Fmax = 40 MHz,   
-period = 25 ns, -waveform {0.000 12.500}, Strategy = ExtraTimingOpt).

Параметр WNS по-прежнему является положительным (отсутствуют TIMING VIOLATIONS) и почти равен 0, следовательно, Fmax для данной конфигурации также равна 40 MHz (так, как при бОльшей частоте и, соответственно, меньшем периоде, будут TIMING VIOLATIONS).

На рисунке 24 продемонстрирована занимаемая площадь. Использовано больше площади устройства, так как конфигурация SCR1\_FAST\_MUL, предоставляет выполнение умножения за 1 цикл, которое реализовано на комбинаторной логике. Она является частью стадии выполнения, которая в свою очередь, реализуется в АЛУ, являющегося частью модуля i\_pipe\_top. Именно в нем и видно основное изменение занимаемой площади.

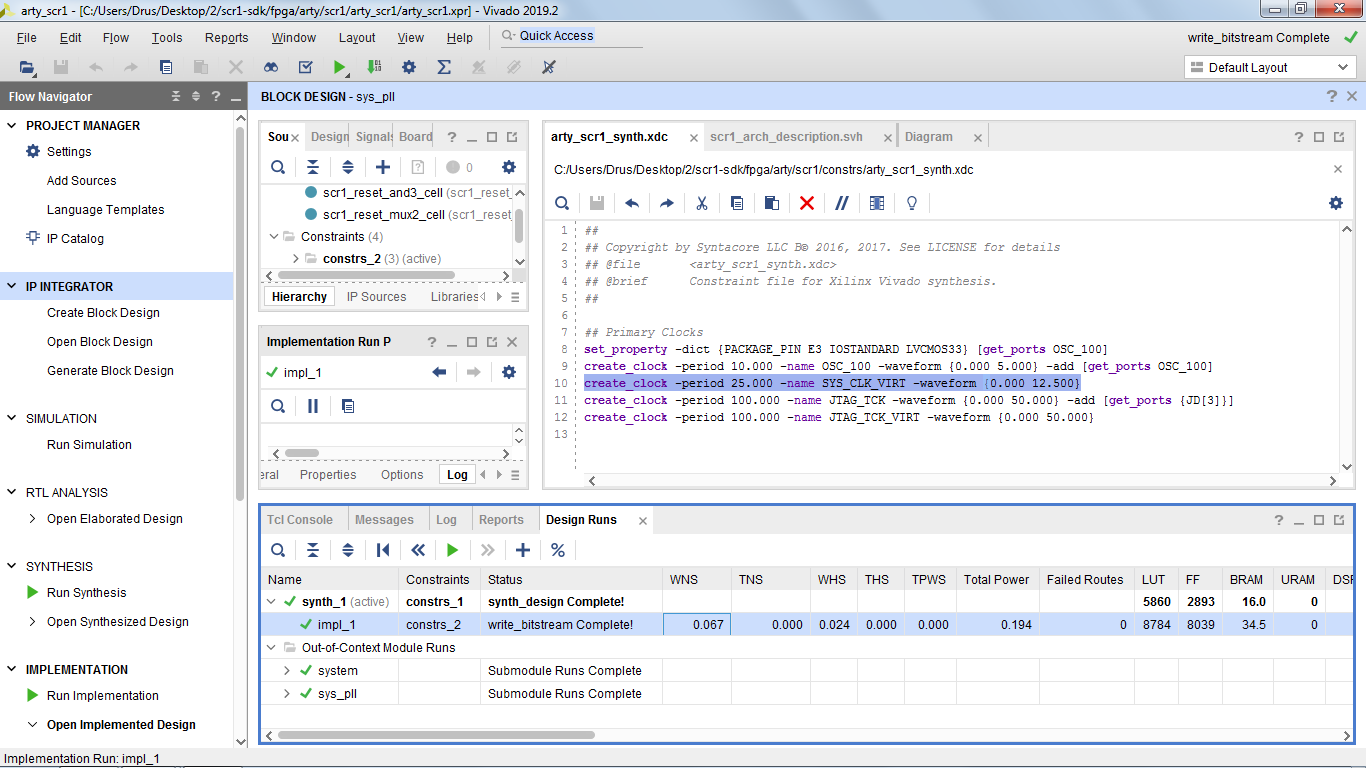


Рисунок 23 Результат выполнения операции Generate Bitstream при тактовой частоте 40 MHz, периодом 25 ns и стратегией реализации ExtraTimingOpt

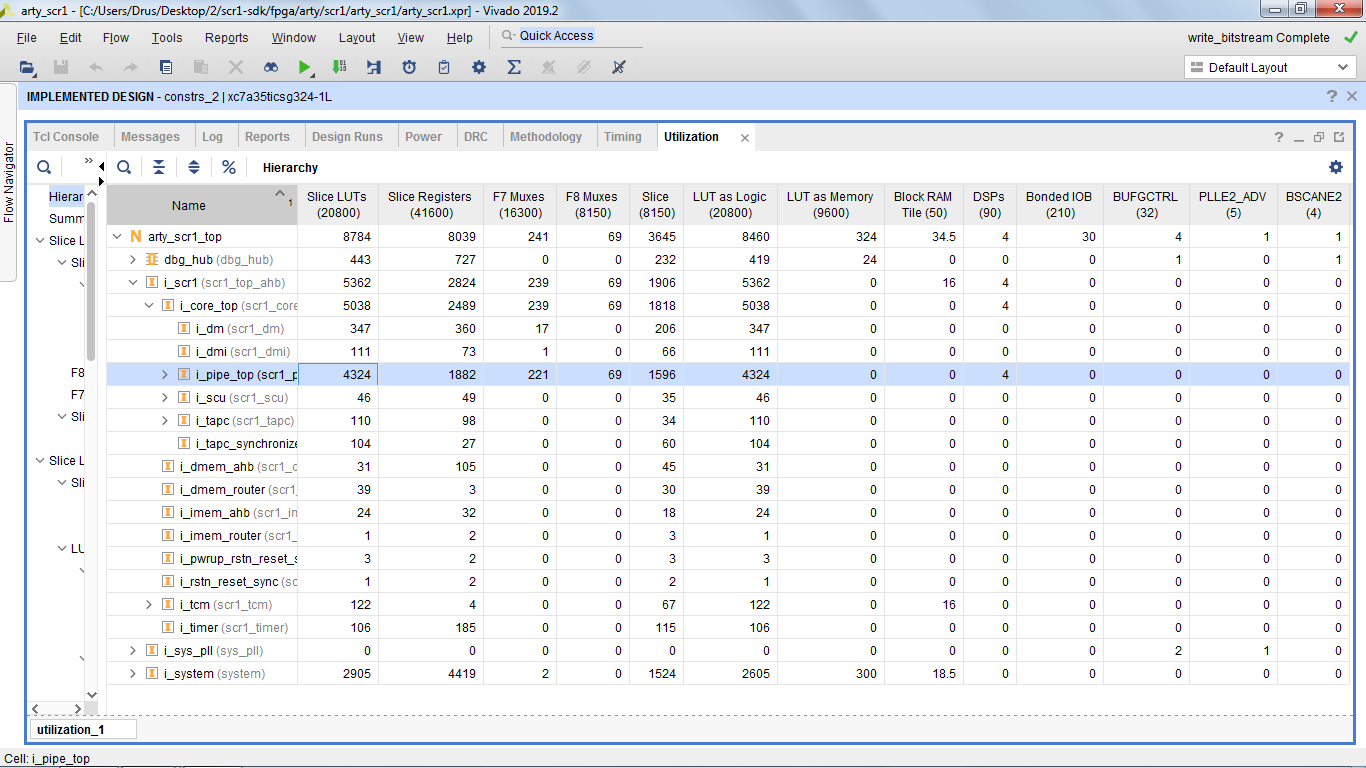


Рисунок 24 Занимаемая площадь конфигурации RVIM +FAST\_MUL

В таблице 3 представлено сравнение по трём конфигурациям.

Таблица 3 Сравнение конфигураций

|  |  |  |  |
| --- | --- | --- | --- |
| Configuration | *Area, LUTs* |  | *Greatest diff in* |
| **IMC\_MAX** | 9190 | 33,(3) | - |
| **RVIM -FAST\_MUL** | 8714 | 40 | i\_scr1->i\_core\_top->i\_pipe\_top |
| **RVIM +FAST\_MUL** | 8784 | 40 | i\_scr1->i\_core\_top->i\_pipe\_top |