Санкт-Петербургский Политехнический Университет Петра Великого

Институт компьютерных наук и кибербезопасности

Высшая школа компьютерных технологий и информационных систем

ОТЧЕТ ПО ЛАБОРАТОРНОЙ РАБОТЕ Lab1\_Z2

Дисциплина: Проектирование реконфигурируемых гибридных вычислительных систем

Тема: Введение в Vivado HLS

Выполнил студент гр. Шеметов С.А.

гр.5140901/21502

Руководитель, доцент Антонов А.П.

«\_\_» \_\_\_\_\_\_ 2023

Санкт-Петербург

2023

Оглавление

[1. Задание 3](#_Toc146400048)

[2. Исходный код функции 3](#_Toc146400049)

[3. Исходный код теста 3](#_Toc146400050)

[3.1. Моделирование 3](#_Toc146400051)

[4. Первое решение — Solution\_1 5](#_Toc146400052)

[4.1. Исходные настройки 5](#_Toc146400053)

[4.2. Синтез 5](#_Toc146400054)

[4.2.1. Оценка временных параметров 5](#_Toc146400055)

[4.2.2. Оценка производительности и аппаратных затрат 5](#_Toc146400056)

[4.2.3. Планировщик Schedule Viewer 6](#_Toc146400057)

[4.3. C/RTL моделирование 7](#_Toc146400058)

[4.3.1. Анализ результатов 7](#_Toc146400059)

[4.3.2. Отчет по производительности 7](#_Toc146400060)

[4.3.3. Временная диаграмма 7](#_Toc146400061)

[5. Второе решение — Solution\_2 8](#_Toc146400062)

[5.1. Исходные настройки 8](#_Toc146400063)

[5.2. Синтез 8](#_Toc146400064)

[5.2.1. Оценка временных параметров 8](#_Toc146400065)

[5.2.2. Оценка производительности и аппаратных затрат 8](#_Toc146400066)

[5.2.3. Планировщик Schedule Viewer 9](#_Toc146400067)

[5.3. C/RTL моделирование 10](#_Toc146400068)

[5.3.1. Анализ результатов 10](#_Toc146400069)

[5.3.2. Отчет по производительности 10](#_Toc146400070)

[5.3.3 Временная диаграмма 10](#_Toc146400071)

[6. Выводы 11](#_Toc146400072)

[6.1. Сравнение параметров двух решений 11](#_Toc146400073)

[6.2. Какое из двух решений имеет меньший II? 12](#_Toc146400074)

[6.3 Какое из двух решений обеспечивает большее быстродействие (в ns)? 12](#_Toc146400075)

[6.4 Какое из двух решений требует меньших аппаратных затрат? 13](#_Toc146400076)

[6.5 Какое из двух решений Вам кажется более оптимальным? 13](#_Toc146400077)

1. Задание

* Разработать на языке С++ описание функции, реализующей следующий алгоритм

outArr[i] = inArr[i] + inA + inB + inC, где i=0, 1, ... (ROWS-1). ROWS=3

* Тип данных для inArr[i], inA, inB, inC, inD – short; для outArr[i] – int;
* Разработать тест, обеспечивающий автоматическую проверку получаемых результатов моделирования разработанной функции.
* Создать, провести исследование и сравнительный анализ двух аппаратных реализаций разработанного на языке С++ описания функции.
  + Микросхема: xa7a12t-csg325-1q
  + clock period 6; clock\_uncertainty 1 (для решения Solution 1)
  + clock period 10; clock\_uncertainty 1 (для решения Solution 2)

2. Исходный код функции

Исходный код синтезируемой функции приведен на рисунке 1.

*Изображение выглядит как текст, программное обеспечение, Мультимедийное программное обеспечение, Графическое программное обеспечение

Автоматически созданное описание*

*Рис.1. Исходный код функции lab1\_z2*

Функция принимает 5 аргументов — 3 скаляра и 2 векторных значения, считает значение по формуле и записывает результат расчета в массив outArr.

3. Исходный код теста

Исходный код теста для проверки функции lab1\_z2, исправленный в соответствии с заданием, приведен на рисунке 2. Тест обеспечивает проверку корректной работы функции.

3.1. Моделирование

Результаты моделирования исходного кода синтезируемой функции приведены на рисунке 3. Результаты моделирования показывают, что тест успешно пройден — так как условия теста выполняются (функция main возвращает 0) и результат расчета совпадает с ожидаемыми значениями.

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

Автоматически созданное описание*

*Рис.2. Исходный код теста lab1\_z2*

*Изображение выглядит как текст, снимок экрана, Шрифт, программное обеспечение

Автоматически созданное описание*

*Рис.3. Лог результата выполнения моделирования*

Для проверки правильности работы теста в него было внесено изменение в ожидаемый результат - expectedArr[i] = inArr [i] + inA + inB - inC. Результат моделирования приведён на рисунке 4. Он показывает, что тест отрабатывает ошибку корректно.

*Изображение выглядит как текст, снимок экрана, программное обеспечение, Шрифт

Автоматически созданное описание*

*Рис.4. Лог результата при ошибке в тесте*

4. Первое решение — Solution\_1

4.1. Исходные настройки

При создании первого решения задаются следующие настройки:

* clock period: 6,
* clock uncertain: 1,
* part: xa7a12tcsg325-1q

4.2. Синтез

4.2.1. Оценка временных параметров

Результаты оценки временных параметров представлены на рисунке 5.

Target timing — заданная требуемая частота равная 6 нс.

Estimated timing — достигнутая частота равна 4.333 нс.

**Изображение выглядит как текст, снимок экрана, Шрифт, число

Автоматически созданное описание**

*Рис.5. Оценка временных параметров для solution\_1*

4.2.2. Оценка производительности и аппаратных затрат

Результаты оценки производительности и аппаратных затрат представлены на рисунке 6.

Latency(cycle) — необходимое количество циклов для получения вывода равно 10 тактов. Latency(absolute) — необходимое количество времени для получения вывода равное 60 нс. Для цикла функции (VITIS\_LOOP\_3\_1) получены следующие данные: Latency(cycle) равен 9 тактам, IL(Iteration latency) равен 3 тактам и Trip Count равен 3.

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

Автоматически созданное описание

*Рис.6. Оценка производительности и аппаратных затрат для solution\_1*

4.2.3. Планировщик Schedule Viewer

Результаты планирования выполнения функции приведены на рисунке 7.

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

Автоматически созданное описание*

*Рис.7. Планировщик для solution\_1*

На рисунке 7 представлено 4 столбца соответствующих 4м тактам. На первом такте происходить чтение переменных *a*, *b* и *c*, а затем сложение *a* и *b* (операция *add\_ln11(+)*). Второй-четвертый такт отображают одну итерацию цикла. В ней читается и инкрементируется переменная *i*,читается переменная *x*, *x* и *c* суммируются (операция *add\_ln11\_1(+)*) и полученные ранее две суммы складываются в переменной *y*.

Количество тактов на итерацию цикла (Iteration Latency) равно 3 тактам, поэтому Loop Latency совпадает с результатами Performance & Resource Estimates и равно 9 тактам (3 итерации \* 3 такта). Function Latency равно 10 тактам (Loop Latency + 1 такт).

4.3. C/RTL моделирование

4.3.1. Анализ результатов

Результаты, приведенные на рисунке 8, показывают, что синтезированное описание работает в соответствии с ожиданием — так как в отчёте моделирования отображается статус Pass.

 *Рис.8. Результат моделирования C/RTL для solution\_1*

4.3.2. Отчет по производительности

Отчет по производительности, полученный при совместном Си и RTL моделировании приведен на рисунке 9. Из отчета следует, что для всего проекта получены те же Latency и II, что и при синтезе, показанные на рисунке 6. Однако стоит отметить, что значения цикла соответствуют общим значениям всего проекта, так как программа отдельно не выделяет их.

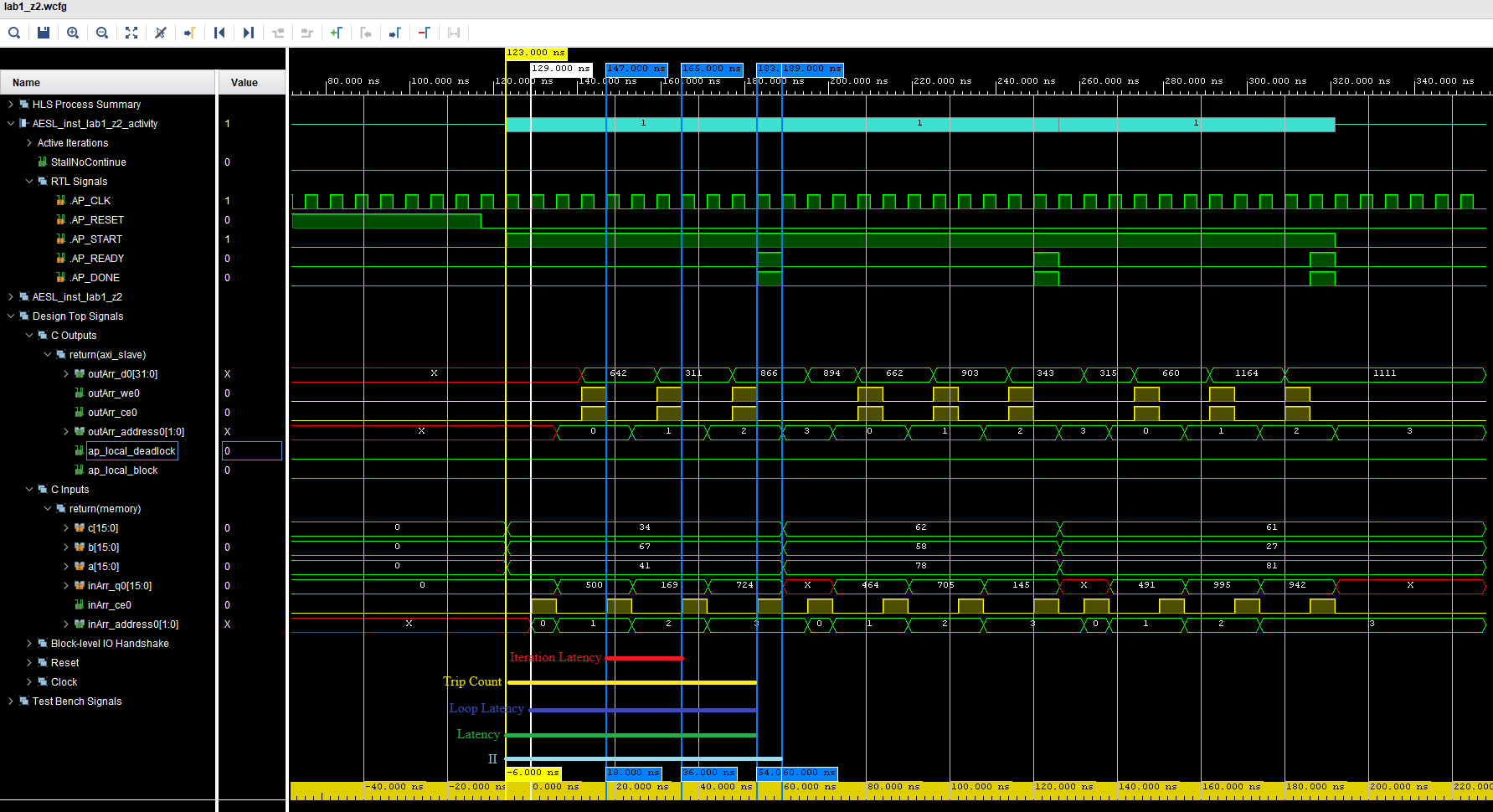
*Изображение выглядит как текст, снимок экрана, программное обеспечение, Мультимедийное программное обеспечение

Автоматически созданное описание*

*Рис.9. Отчет по производительности C/RTL для solution\_1*

4.3.3. Временная диаграмма

Временная диаграмма совместного C/RTL моделирования приведены на рисунке 10. На временной диаграмме показаны сигналы Disign Top Signals и размечены Iteration Latency (красный) = 3 тактам, Trip Count (желтый) = 3, Loop Latency (темно-синий) = 9 тактам, Function Latency (зеленый) = 10 тактам, Initiation Interval (голубой) = 11 тактам.

*Рис.11. Временная диаграмма C/RTL для solution\_1*

5. Второе решение — Solution\_2

5.1. Исходные настройки

При создании второго решения задаются следующие настройки:

* clock period: 10,
* clock uncertain: 1,
* part: xa7a12tcsg325-1q

5.2. Синтез

5.2.1. Оценка временных параметров

Результаты оценки временных параметров представлены на рисунке 12.

Из рисунка видно, что заданная ожидаемая частота (Target) равно 10 нс, а достигнутая (Estimated) равна 8.631 нс.

Изображение выглядит как текст, снимок экрана, Шрифт, число

Автоматически созданное описание

*Рис.12. Оценка временных параметров для solution\_2*

5.2.2. Оценка производительности и аппаратных затрат

Результаты оценки производительности и аппаратных затрат представлены на рисунке 13.

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

Автоматически созданное описаниеРис.13. Оценка производительности и аппаратных затрат для solution\_2*

Из рисунка видно, что Latency(cycle) равно 7 тактам и Latency(absolute) 70 нс. Для цикла функции (VITIS\_LOOP\_3\_1) получены следующие данные: Latency(cycle) равен 6 тактам, IL(Iteration latency) равен 2 тактам и Trip Count равен 3.

5.2.3. Планировщик Schedule Viewer

Результаты планирования выполнения функции приведены на рисунке 14.

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

Автоматически созданное описание*

*Рис.14. Планировщик для solution\_2*

На рисунке 14 представлено 3 столбца соответствующих 3м тактам. На первом такте как и в Solution1 происходить чтение переменных *a*, *b* и *c*, а затем сложение *a* и *b*. Второй-третий такт отображают одну итерацию цикла и выполняют все остальные операции.

Количество тактов на итерацию цикла (Iteration Latency) равно 2 тактам, поэтому Loop Latency совпадает с результатами Performance & Resource Estimates и равно 6 тактам (3 итерации \* 2 такта). Function Latency равно 7 тактам (Loop Latency + 1 такт).

5.3. C/RTL моделирование

5.3.1. Анализ результатов

Результаты, приведенные на рисунке 15, показывают, что синтезированное описание работает в соответствии с ожиданием — так как в отчёте моделирования отображается статус Pass.



*Рис.15. Результат моделирования C/RTL для solution\_2*

5.3.2. Отчет по производительности

Отчет по производительности, полученный при совместном Си и RTL моделировании приведен на рисунке 16. Из отчета следует, что для всего проекта получены те же Latency и II, что и при синтезе, показанные на рисунке 13. Как и в Solution1 значения цикла соответствуют общим значениям всего проекта, так как программа отдельно не выделяет их.

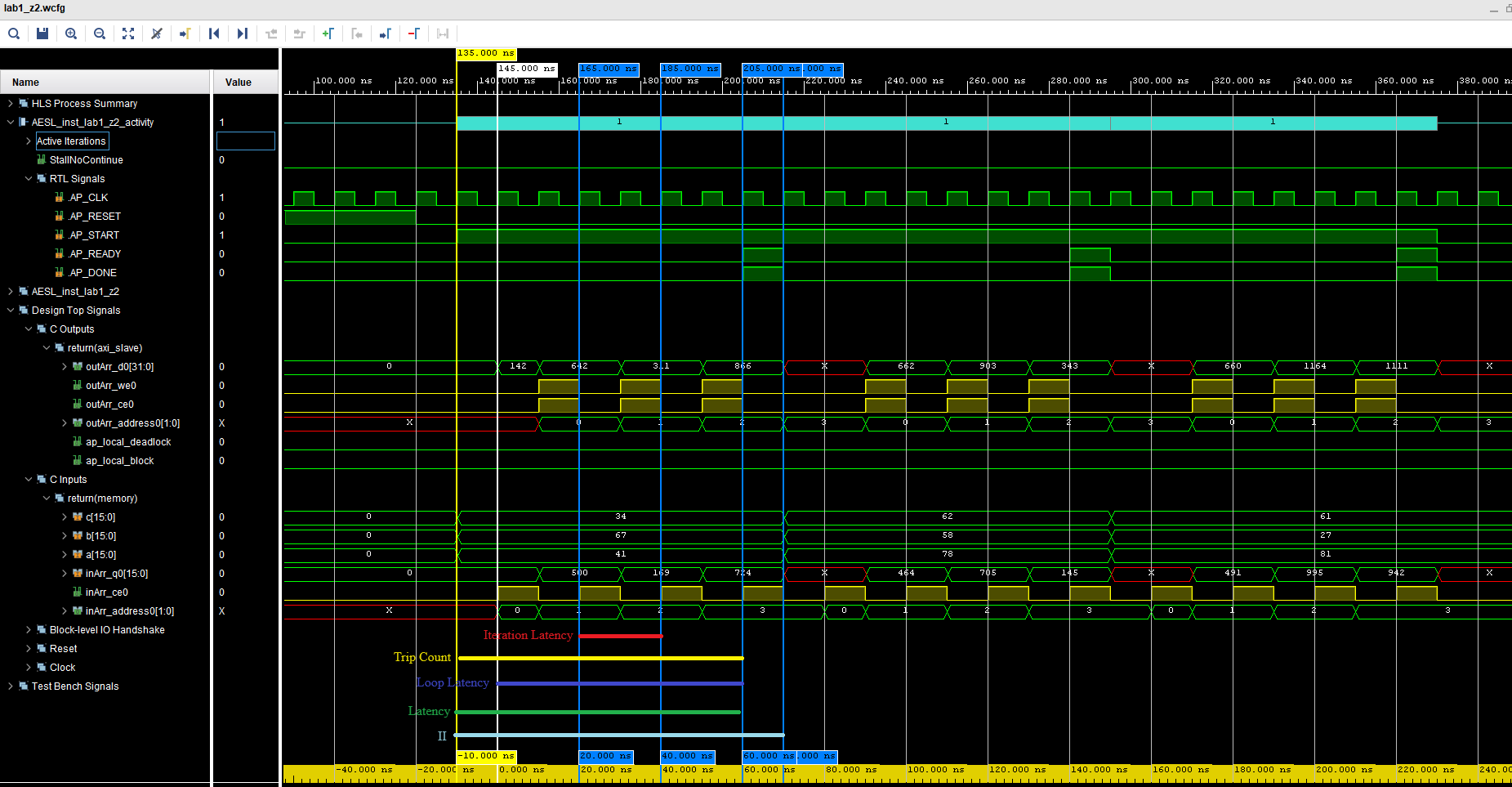
Изображение выглядит как текст, снимок экрана, программное обеспечение, Мультимедийное программное обеспечение

Автоматически созданное описание

*Рис.16. Отчет по производительности C/RTL для solution\_2*

5.3.3 Временная диаграмма

Временная диаграмма совместного C/RTL моделирования приведены на рисунке 17. На временной диаграмме показаны сигналы Disign Top Signals и размечены Iteration Latency (красный) = 2 тактам, Trip Count (желтый) = 3, Loop Latency (темно-синий) = 6 тактам, Function Latency (зеленый) = 7 тактам, Initiation Interval (голубой) = 8 тактам.

*Рис.17. Временная диаграмма C/RTL для solution\_2*

6. Выводы

6.1. Сравнение параметров двух решений

В данной работе были реализованы два решения с разной частотой, которые могут отличаться в зависимости от задаваемого параметра. Результаты сравнения параметров двух решений приведены на рисунке 18.

Изображение выглядит как текст, снимок экрана, Шрифт, число

Автоматически созданное описание

*Рис.18. Сравнение двух решений*

Как видно из рисунка разница в задаваемой частоте влияет на большинство параметров. Во втором решений количество тактов, требуемых для вычислений, стало меньше, как и количество аппаратных ресурсов. А оценка времени выполнения для функции во втором случае больше, как и время итерации цикла для второго решения (Latency(absolute)).

6.2. Какое из двух решений имеет меньший II?

Первое решение II1 = 11 тактов, второе решение имеет II2 = 8 тактов, тогда II1 > II2. При переводе II в наносекунды II1 = 11 ∙ 4.333 = 47.663 нс и II2 = 8 ∙ 8.631 = 69.048 нс. Следовательно второе решение имеет большее значение II в наносекундах и меньшее в тактах.

6.3 Какое из двух решений обеспечивает большее быстродействие (в ns)?

Наибольшее быстродействие обеспечивает первое решение, так как его Latency(absolute) меньше, чем у второго решения на 10 нс.

6.4 Какое из двух решений требует меньших аппаратных затрат?

Первое решение требует большее количество аппаратных ресурсов. Количество триггеров FF в первом решений требует на 18 штук больше, чем во втором, а количество LUT превышает на 4 штуки.

6.5 Какое из двух решений Вам кажется более оптимальным?

На мой взгляд, нет более оптимального решения в каком-то из этих двух решений, так как мы получаем, что, первое решение имеет большее быстродействие, но затрачивает большее количество аппаратных ресурсов (в особенности FF триггеров), чем второе. Соотношение быстродействие/ресурсы будет зависит от конкретно поставленной задачи.