

Задание lab6 (4 часа лабораторных занятий)

□ Часть lab6_1

- На языке Verilog создайте:
- Функцию, обеспечивающую преобразование двоичного кода в код Грея и обратное преобразование кода Грея в двоичный код:

- ✓ Особенности:

- Разрядность преобразуемого кода задается параметром WIDTH
- Направление преобразования задается параметром DIR
 - 0: двоичный код в код Грея
 - 1: код Грея в двоичный
- необходимо использовать циклы

□ Модуль

- ✓ параметризованный, параметр W - разрядность преобразуемого кода
- ✓ содержащий два экземпляра разработанной функции:
 - Один экземпляр функции преобразует входной двоичный код в код Грея (разрядность задается параметром W)
 - Второй экземпляр функции преобразует код Грея в выходной двоичный код (разрядность задается параметром W)

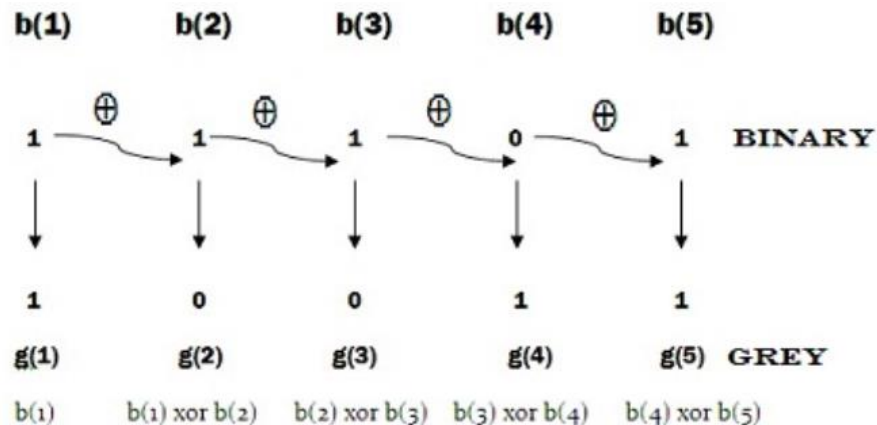
Часть lab6_1 продолжение описания

- ❑ Надо осуществить моделирование модуля при значении параметра $W=4$ (разрядность преобразуемых кодов – 4)
- ❑ Реализовать модуль в СБИС ПЛ и проверить работу на плате при значении параметра $W=4$ (разрядность преобразуемых кодов – 4)
- ❑ Входной двоичный код поступает с переключателей $SW[3:0]$
 - ✓ Полученный (с помощью первого экземпляра функции) код Грея отображается на светодиодах $led[7:4]$ и поступает на второй экземпляр функции для преобразования в двоичный код.
 - ✓ Полученный (с помощью второго экземпляра функции) двоичный код отображается на светодиодах $led[3:0]$
- ❑ Дополнительные требования:
 - ✓ стандарты и номера выводов СБИС для платы miniDiLaB_CIV задайте с помощью атрибутов.
- ❑ Имя проекта – lab6_1. Имя модуля верхнего уровня – lab6_1.

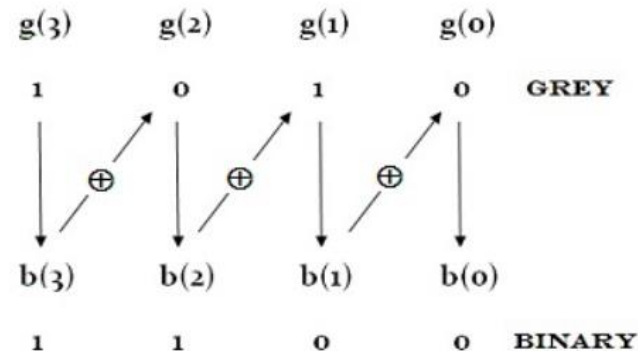
Grey Code to Binary Conversion

Binary to Grey Code Conversion

Convert the binary 11101_2 to its equivalent Grey code



Convert the Grey code 1010 to its equivalent Binary



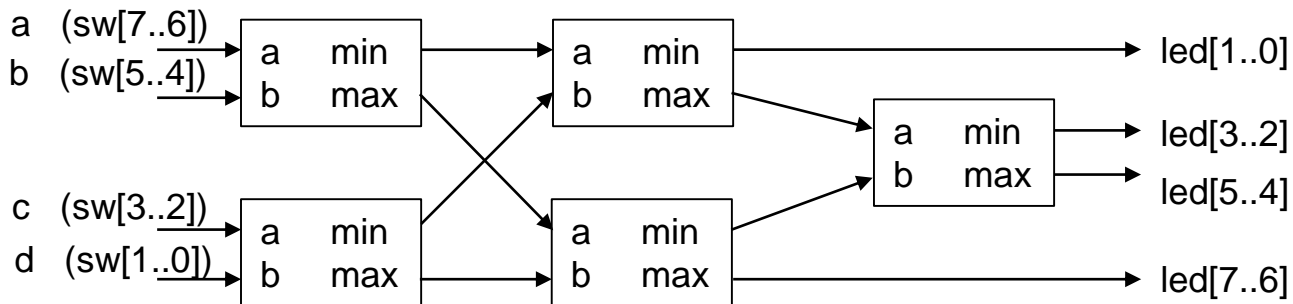
i.e

$$b(3) = g(3)$$
$$b(2) = b(3) \oplus g(2)$$
$$b(1) = b(2) \oplus g(1)$$
$$b(0) = b(1) \oplus g(0)$$

Часть lab6_2

- ❑ На языке Verilog создайте описание:
- ❑ Task (задачи) сортировки двух чисел (комбинационная схема)
 - ✓ Передаваемые значений – два числа a и b
 - ✓ Возвращаемые значения: min – меньшее из a и b; max – большее из a и b.
- ❑ Устройства сортировки четырех 2-разрядных чисел, использующего созданную задачу. На входах и выходах устройства должны быть регистры
- ❑ Дополнительные требования:
 - ✓ стандарты и номера выводов СБИС для платы miniDiLaB_CIV задайте с помощью атрибутов.
- ❑ Имя проекта – lab6_2. Имя модуля верхнего уровня – lab6_2
- ❑ Выводы устройства:
- ❑ Входы:
 - ✓ Переключатель sw[7:6] – операнд A
 - ✓ Переключатель sw[5:4] – операнд B
 - ✓ Переключатель sw[3:2] – операнд C
 - ✓ Переключатель sw[1:0] – операнд D
 - ✓ Clk – тактовый вход
 - ✓ Rst (кнопка pbA) – вход синхронного сброса
- ❑ Выходы – операнды, отсортированные в по убыванию или по возрастанию
 - ✓ светодиоды led[7:6]
 - ✓ светодиоды led[5:4]
 - ✓ светодиоды led[3:2]
 - ✓ светодиоды led[1:0]

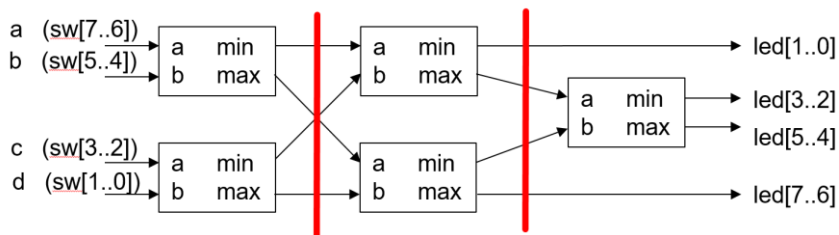
❑ Сортировка



Часть lab6_3

❑ Осуществите конвейеризацию устройства из части Lab6_2

- ✓ На рисунке показано место для регистров конвейеризации.



❑ Дополнительные требования:

- ✓ стандарты и номера выводов СБИС для платы miniDiLaB_CIV задайте с помощью атрибутов.

✓ Проведите исследование:

- Для исходной (lab6_2) реализации зафиксируйте
 - с помощью RTL Viewer структуру.
 - Аппаратные затраты
 - Максимальную тактовую частоту
- Для конвейерной реализации зафиксируйте результат
 - с помощью RTL Viewer структуру.
 - Аппаратные затраты
 - Максимальную тактовую частоту
- Осуществите сравнение

✓ Для конвейерной реализации осуществите моделирование.

✓ Осуществите проверку на плате

- ❑ Имя проекта – lab6_3. Имя модуля верхнего уровня – lab6_3.

Часть lab6_4

Часть lab6_4

- На языке Verilog введите представленный ниже текст

```
module lab7_ (sel, x, y, z, q1, q2, q3);  
input [2:0] sel;  
input x, y, z;  
output reg q1, q2, q3;  
always @*  
begin  
    if (sel == 3'h1) q1 = x;  
    else if (sel == 3'h2) q2 = y;  
        else if ( sel == 3'h3) q3 = z;  
        else  
            begin  
                q1 = 1'b0;  
                q2 = 1'b0;  
                q3 = 1'b0;  
            end  
    end  
end  
endmodule
```

- Осуществите анализ и синтез, посмотрите:

- ✓ Предупреждения в процессоре сообщений и проверить наличие предупреждений о триггерах защелках (Latch)
- ✓ синтезированную пакетом Q схему (RTL Viewer), найти триггеры защелки

- Оцените аппаратные затраты (запомните кол-во использованных логических элементов)
- Исправьте описание так, чтобы не нарушая логику работы исключить возможность появления триггеров-защелок.
- Повторите анализ и синтез, убедитесь, что триггеры-защелки отсутствуют (нет предупреждений в процессоре сообщений; результаты синтеза в RTL Viewer)
- Оцените аппаратные затраты (сравните с результатами, полученными ранее)
- Имя проекта – lab6_4. Имя модуля верхнего уровня – lab6_4.