|  |  |  |
| --- | --- | --- |
|  |  |  |
| МИНОБРНАУКИ РОССИИ | | |
| Федеральное государственное бюджетное образовательное учреждение  высшего образования  **«МИРЭА – Российский технологический университет»**  **РТУ МИРЭА** | | |
| Институт Информационных Технологий | | |
| Кафедра Вычислительной техники | | |

|  |  |
| --- | --- |
| **ОТЧЕТ О ВЫПОЛНЕНИИ ПРАКТИЧЕСКИХ РАБОТ** | |
|  | |
| **по дисциплине** | |
| **«**Архитектура вычислительных машин и систем**»** | |
|  | |
| Выполнил студент группы ИМБО-02-22 | *Ким К.С.* |
| Принял старший преподаватель | *Боронников А.С.* |

|  |  |  |
| --- | --- | --- |
| Практические работы выполнены | «\_\_»\_\_\_\_\_\_\_2023 г. |  |
|  |  |  |
| «Зачтено» | «\_\_»\_\_\_\_\_\_\_2023 г. |  |

**Оглавление**

[**Практическая работа №1** 3](#_Toc153286737)

[**Практическая работа №2** 5](#_Toc153286738)

[**Практическая работа №3** 7](#_Toc153286739)

[**Практическая работа №4** 9](#_Toc153286740)

[**Практическая работа №5** 10](#_Toc153286741)

[**Практическая работа №6** 12](#_Toc153286742)

[**Практическая работа №7** 14](#_Toc153286743)

[**Практическая работа №8** 15](#_Toc153286744)

[**Практическая работа №9** 17](#_Toc153286745)

[**Практическая работа №10** 20](#_Toc153286746)

[**Практическая работа №11** 24](#_Toc153286747)

[**СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ** 27](#_Toc153286748)

# **Практическая работа №1**

Задание:

Нужно расписать управляющее слово ЦП для различных микроопераций.

Управляющее слово ВМ составляется из управляющих разрядов MSA, MSB, АЛУ, DS, MSM, WR.

Задача устройства управления – обеспечить выдачу управляющего слова ЦП. Иначе он называется микрокод, который управляет работой аппаратуры ЦП. Обычно он записывается в постоянное запоминающее устройство (ПЗУ) центрального процессора (прошивается).

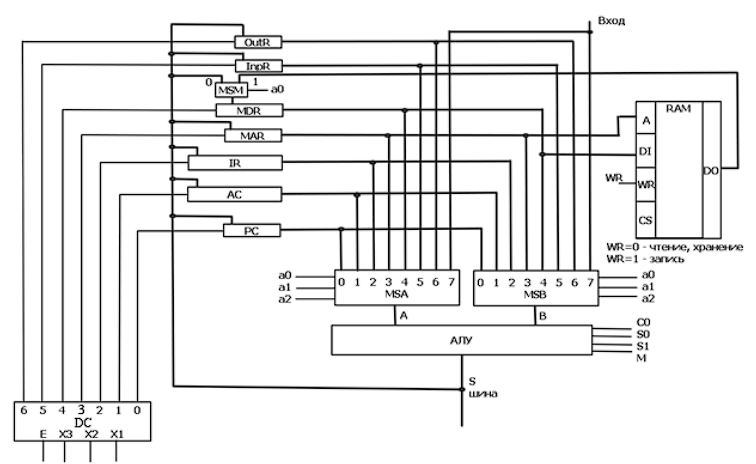


Рисунок 1 – Схема операционного устройства

Управляющее слово: INPR ← IR + MAR

Решение:

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| MSA | | | MSB | | | АЛУ | | | | DC | | | | MSM | RAM |
| 0 | 1 | 0 | 0 | 1 | 1 | 0 | 0 | 1 | 1 | 0 | 1 | 0 | 1 | x | 0 |

# **Практическая работа №2**

Задание:

Составить микрооперации каждого такта циклов выборки и выполнения команды.

Последовательность работы устройства управления задается циклами.

Для начала рассмотрим цикл выборки команды. Ведь прежде, чем выполнить команду, надо выбрать её из памяти ВМ.

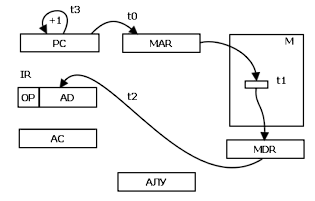


Рисунок 2 – Цикл выборки команды

Для выборки команды из памяти ВМ надо, во-первых, поместить адрес команды из регистра адреса команды в регистр адреса памяти, во-вторых, прочитать команду из ячейки памяти в регистр данных памяти, в-третьих. Переслать команду в регистр команды, в-четвертых, подготовить счетчик команд для выборки следующей команды – инкременировать PC.

То есть, последовательно надо выполнить 4 действия, обозначенные на рис. 12 t0, t1, t2, t3. Следовательно, цикл выборки команды осуществляется за 4 такта.

Само выполнение команды происходит следующим образом:

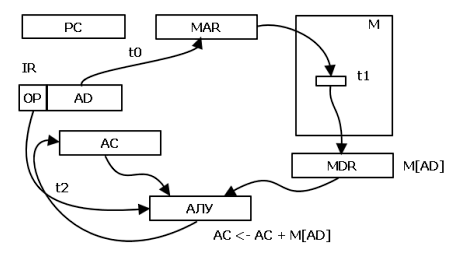


Рисунок 3 – Цикл выполнения команды

Таким образом, мы получили наличие в УУ ЦП следующих циклов:

1. Цикл выборки команды – С0;
2. Цикл дешифрации команды – С1;
3. Цикл выполнения команды – С2;
4. Цикл обработки прерывания – С3.

Команда: AC←M[AD] - 1

Решение:

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | Команда | MSA | MSB | АЛУ | DC | MSM | WR | FR |
| C0t0 | MAR←PC | 000 | xxx | 0000 | 0011 | x | 0 | 00 |
| C0t1 | MDR←M[MAR] | xxx | xxx | xxxx | 0100 | 1 | 0 | 00 |
| C0t2 | IR←MDR | 100 | xxx | 0000 | 0010 | x | 0 | 00 |
| C0t3 | PC←PC + 1; F = 1 | 000 | xxx | 0001 | 0000 | x | 0 | 10 |
| q2C2t0 | MAR←IR[AD] | 010 | xxx | 0000 | 0011 | x | 0 | 10 |
| q2C2t0 | MDR←M[MAR] | xxx | xxx | xxxx | 0100 | 1 | 0 | 10 |
| q2C2t0 | AC←MDR - 1 | 001 | xxx | 1010 | 0001 | x | 0 | 10 |
| q2C2t0 | F=0 | xxx | xxx | xxxx | 1xxx | x | 0 | 00 |

# **Практическая работа №3**

Задание:

1. Определяем исходное состояние регистров.
2. Выписываем микрокоманды циклов выборки команды и выполнения команды.
3. Определяем значения регистров после выполнения каждой микрокоманды.
4. Результаты оформляем в виде таблицы.

Вариант:

1. Записать микрооперации цикла выборки и выполнения команды DEC.
2. По адресу 2E5 записана команда DEC с адресом 379. По этому адресу команды записан операнд 6A4B. В регистре-аккумуляторе находится операнд 56DC.

Определить информацию, которая будет записана в регистрах PC, MAR, MDR, IP, AC после выполнения этой команды.

Решение:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Регистры | PC | MAR | MDR | IR | AC |
| Исх. состояние рег. | 2E5 |  |  |  | 56DC |
| C0t0 :  MAR←PC |  | 2E5 |  |  |  |
| C0t1 :  MDR←M[MAR] |  |  | A379 |  |  |
| C0t2 :  IR←MDR |  |  |  | A379 |  |
| C0t3 :  PC←PC + 1; F = 1 | 2E6 |  |  |  |  |
| q2C2t0 :  MAR←IR[AD] |  | 379 |  |  |  |
| q2C2t1 :  MDR←M[MAR] |  |  | 6A4B |  |  |
| q2C2t2 :  AC←MDR - 1 |  |  |  |  | 6A4A |
| q2C2t3 :  F←0 |  |  |  |  |  |
| Результат: | 2E6 | 379 | 6A4B | A379 | 6A4A |

# **Практическая работа №4**

Задание:

Виртуальная память имеет объем 32 М байт, физическая – 32 к байт, страница - 4 к байт. Слово из виртуальной памяти со страницы № 24 со смещением 99 переписывается на страницу № 6 физической памяти. Записать виртуальный и физический адреса слова в двоичном коде, разделив страницу и смещение.

Решение:

Объем виртуальной памяти = 32 Мбайт = 25 \* 220 байт

Объем физической памяти = 32 Кбайт = 25 \* 215 байт

Размер страницы = 4 Кбайт = 22 \* 210 = 212 байт

Номер виртуальной страницы = 24

Смещение = 99

Номер физической страницы = 6

Виртуальный адрес

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Номер виртуальной страницы | | | | | | | | Смещение | | | | | | | | | | | |
| 0 | 0 | 0 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 0 | 0 | 0 | 1 | 1 |

Физический адрес

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Номер физической страницы | | | Смещение | | | | | | | | | | | |
| 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 0 | 0 | 0 | 1 | 1 |

# **Практическая работа №5**

Задание:

К компьютеру подключены 4 периферийных устройства (ПУ) с номерами 0, 1, 2 и 3 (в порядке убывание их приоритета). Вектор-адреса (VAD) этих устройств располагаются в оперативной памяти (ОП) по адресам 00, 01, 10, 11, а соответствующие драйвера, т.е. программы обслуживания прерываний (ПОП), находятся по адресам 101-200, 201-300, 301-400 и 401-500.

Объектный код прикладной программы, исполнение которой прерывается при обращениях к ПУ, хранятся в ОП в области адресов 701-1500.

Для сохранения адресов возврата при прерываниях предусмотрен стек.

В соответствии с заданием, где конкретизированы название ПУ и приведен порядок прерываний прикладной программы от 2-х из перечисленных ПУ, привести поясняющий рисунок, где следует схематически изобразить стек, оперативную память с перечисленными адресами для VAD, для программ обслуживания прерываний и прикладной программы, а также указать с помощью стрелок порядок выполнения вычислительного процесса в ходе прерываний. Чтобы порядок выполнения был более ясным, возле стрелок следует поставить последовательные номера: 0,1,2,3 и т.д.

ПУ: 0. Диск 1. Принтер 2. Манипулятор 3. Тачпад.

При выполнении команды 876 прикладной программы пришло прерывание от манипулятора. Далее, при выполнении команды 335 ПОП манипулятора пришло прерывание от диска.

Решение:

Изобразить на рисунке ход выполнения вычислительного процесса

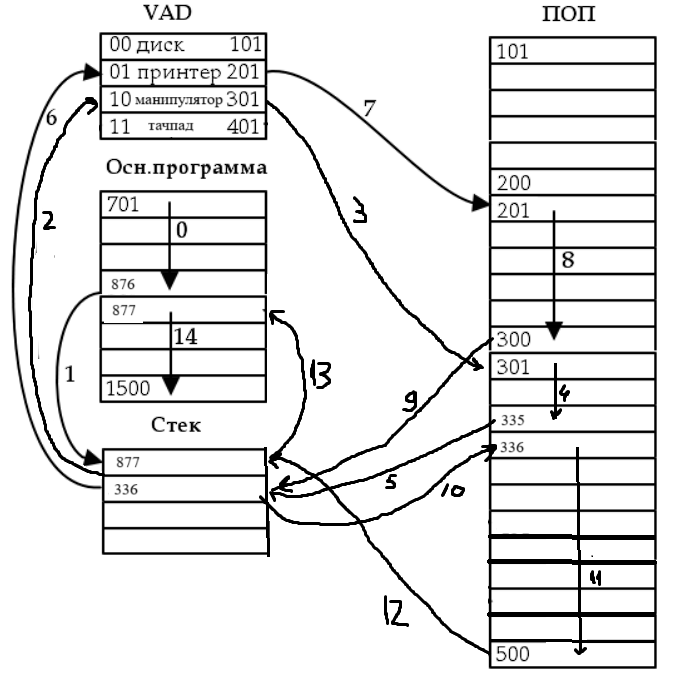


Рисунок 4 – Последовательность передачи управления вычислительным процессом при обработке прерываний от внешних устройств.

# **Практическая работа №6**

Задание:

1. Какие типы ЗУ различают по способу выборки информации?
2. Что такое сверхоперативная память?
3. В чем состоит проблема согласования пропускной способности процессора и памяти

Преобразовать в ПОЛИЗ арифметическое выражение Q=А/(B+C)/(D-E).

Изобразить на рисунке ход вычислительного процесса в стековой ВМ для данного примера, а также в виде таблицы последовательность выполнения арифметических операций при значениях А=12, В=1, С=3, D=5, E=2.

Решение:

1. Типы бывают статистические и динамические.
2. Быстродействующая память, расположенное между процессором и ОЗУ
3. Тактовая частота работы процессора значительно выше, чем тактовая частота, чем тактовая частота ОЗУ, процессор «простаивает», ожидая данные.

Q=А/(B+C)/(D−E)

Преобразуется в формулу

Q=ABC+/DE−/

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Шаг | Формула | Левый знак | Операнды | Результат | Новая формула |
| 1 | 12 1 3 + / 5 2 − / | + | 1 и 3 | 4 | 12 4 / 5 2 − / |
| 2 | 12 4 / 5 2 − / | − | 5 и 2 | 3 | 12 4 / 3 / |
| 3 | 12 4 / 3 / | / | 12 и 4 | 3 | 3 3 / |
| 4 | 3 3 / | / | 3 и 3 | 1 | 1 |

На рисунке 5 изображен расчет выражения с использованием стека.

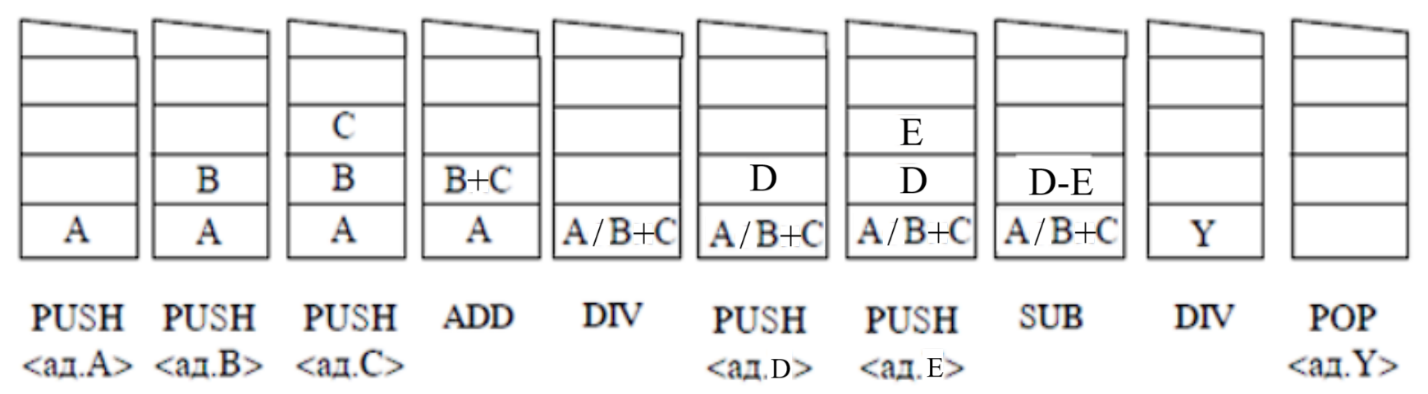


Рисунок 5 – Пример использования стека для вычисления выражений.

# **Практическая работа №7**

Задание:

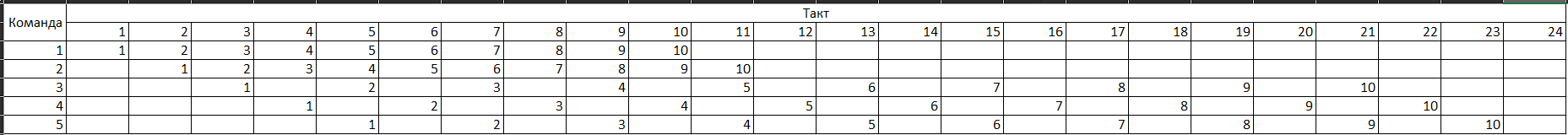
1. Конвейерная обработка в процессоре составляет 9 этапов, каждый длительностью в 1 такт. Определите среднюю длительность выполнения одной команды при количестве команд 12 и отсутствии конфликтов в конвейере.
2. Конвейерная обработка в процессоре составляет 5 этапов, каждый длительностью в 1 такт, кроме 3-го этапа, который составляет 2 такта. Определите среднюю длительность выполнения одной команды при количестве команд 10 и отсутствии конфликтов в конвейере

Решение:

1. По формуле TNK = (K + (N – 1)) \* TK

Средняя длительность выполнения одной команды =

= ((9 + (12 – 1)) \* 1) / 12 ≈ 1.66

1. Построим таблицу

Средняя длительность выполнения одной команды =

= 24 / 10 = 2.4

# **Практическая работа №8**

Задание:

Интернет-провайдер назначил компьютеру IP-адрес: 139.210.75.87/25 (после косой черты за IP-адресом указано число разрядов маски сети провайдера). Количество подсетей провайдера – 4.

Записать IP-адрес в двоичном коде, подчеркнуть биты блока IP-адреса сети провайдера (одной чертой) и биты адресов подсетей (двумя чертами);

Записать в двоичном виде блок адресов сети провайдера;

Записать в дот-нотации блок адресов сети провайдера;

Записать в двоичном виде маску сети;

Записать в дот-нотации маску сети;

Определить номер подсети, в которой находится компьютер (в десятичной системе);

Определить номер компьютера в подсети (в десятичной системе);

Определить максимальное количество компьютеров в подсети.

(Для каждого студента цифровые величины в задании индивидуальны)

Решение:

IP: 139.210.75.87/25

Количество подсетей: 4 = 22

IP-адрес в двоичной системе: 10001011.11010011.01001011.01010111

Дот-нотации блок адресов сети провайдера:

Маска сети: 1111111.11111111.11111111.10000000

Дот-нотации маску сети: 255.255.255.128

Умножаем IP-адрес на маску сети

Номер подсети, котором находится ПК: 2 (выделено желтым цветом)

Номер ПК в подсети: 23 (выделено зелёным цветом)

Максимальное количество компьютеров в подсети: 25 = 32

# **Практическая работа №9**

Задание:

* 1. Изучить правила построения, принцип работы логических схем.
  2. Синтезировать электрическую принципиальную схему логического устройства, описанного заданным преподавателем уравнением в алгебраической форме.
  3. Нарисовать синтезированную схему в графическом редакторе САПР QUARTUS II.
  4. Произвести симуляцию работы схемы. Зарисовать диаграммы работы и по ее результатам заполнить таблицу истинности смоделированной схемы.

Решение:

Данное решение представлено на Листинге – 1 и на Листинге – 2.

*Листинг 1 – lab1var10.v*

|  |
| --- |
| module lab1var10(  input A, B, C, D,  output Y  );  assign Y = (B ^ (A \* C)) + ~(A & D) + (~B) & C;  endmodule |

*Листинг 2 – test.v*

|  |
| --- |
| module test;  reg A\_in, B\_in, C\_in, D\_in;  wire Y\_out;  lab1var10 uut(  .A(A\_in),  .B(B\_in),  .C(C\_in),  .D(D\_in),  .Y(Y\_out)  );  initial begin  A\_in = 1'b0;  B\_in = 1'b0; |

*Продолжение Листинга 2*

|  |
| --- |
| C\_in = 1'b0;  D\_in = 1'b0;  #20;  A\_in = 1'b0;  B\_in = 1'b0;  C\_in = 1'b0;  D\_in = 1'b1;  #20;  A\_in = 1'b0;  B\_in = 1'b0;  C\_in = 1'b1;  D\_in = 1'b0;  #20;  A\_in = 1'b0;  B\_in = 1'b0;  C\_in = 1'b1;  D\_in = 1'b1;  #20;  A\_in = 1'b0;  B\_in = 1'b1;  C\_in = 1'b0;  D\_in = 1'b0;  #20;  A\_in = 1'b0;  B\_in = 1'b1;  C\_in = 1'b0;  D\_in = 1'b1;  #20;  A\_in = 1'b0;  B\_in = 1'b1;  C\_in = 1'b1;  D\_in = 1'b0;  #20;  A\_in = 1'b0;  B\_in = 1'b1;  C\_in = 1'b1;  D\_in = 1'b1;  #20;  A\_in = 1'b1;  B\_in = 1'b0;  C\_in = 1'b0;  D\_in = 1'b0;  #20;  A\_in = 1'b1;  B\_in = 1'b0;  C\_in = 1'b0;  D\_in = 1'b1;  #20;  A\_in = 1'b1;  B\_in = 1'b0;  C\_in = 1'b1;  D\_in = 1'b0;  #20;  A\_in = 1'b1;  B\_in = 1'b0;  C\_in = 1'b1;  D\_in = 1'b1;  #20;  A\_in = 1'b1;  B\_in = 1'b1;  C\_in = 1'b0; |

*Продолжение Листинга 2*

|  |
| --- |
| D\_in = 1'b0;  #20;  A\_in = 1'b1;  B\_in = 1'b1;  C\_in = 1'b0;  D\_in = 1'b1;  #20;  A\_in = 1'b1;  B\_in = 1'b1;  C\_in = 1'b1;  D\_in = 1'b0;  #20;  A\_in = 1'b1;  B\_in = 1'b1;  C\_in = 1'b1;  D\_in = 1'b1;  #20;  $stop;  end  endmodule |

Результат выполнения программы, показано на рисунке 6:

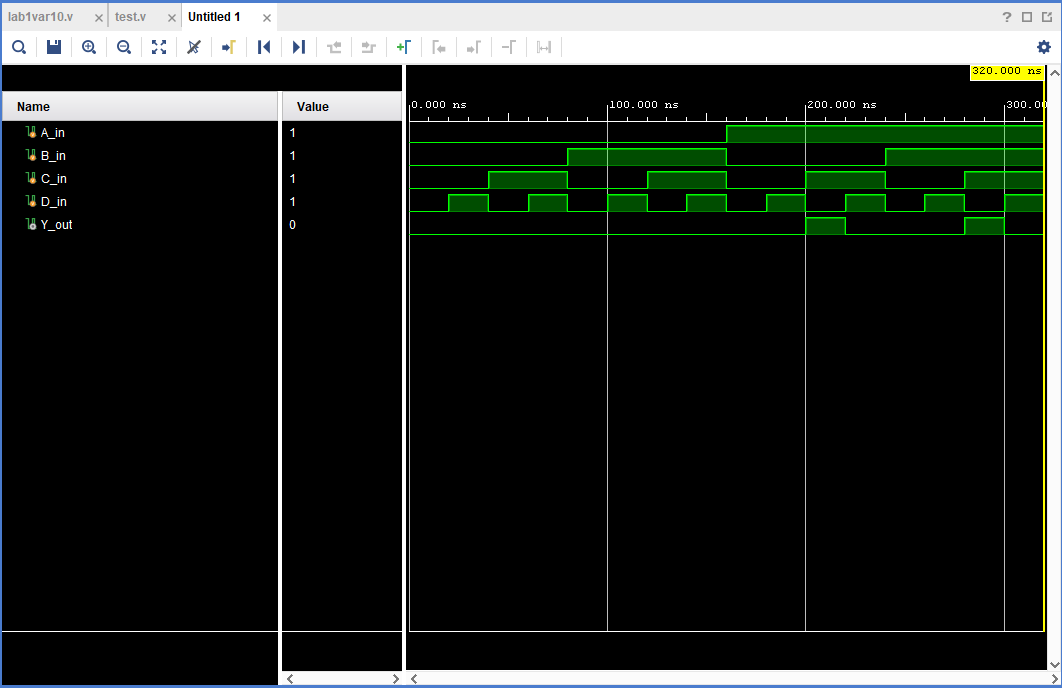


Рисунок 6 – Результат лабораторной работы №1

# **Практическая работа №10**

Задание:

Реализовать модули: дешифратор (DC2\_4.v), приоритетный шифратор (PRCOD4\_2.v: по положению старшей «1» на входе) и 2-разрядный регистр (RG\_2.v: сброс асинхронный по высокому уровню сигнала). Выполнить их коммутацию в TOP-модуле, согласно рисунку 7:

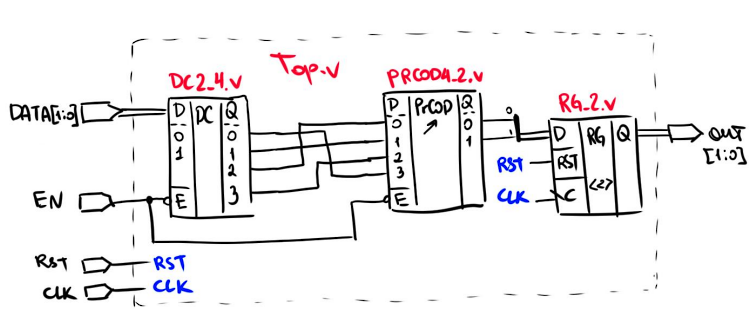


Рисунок 7 – TOP-модуль

Решение:

Данное решение представлено на Листинге – 1, на Листинге – 2, Листинге – 3, Листинге – 4 и на Листинге – 5.

*Листинг 1 – DC2\_4.v*

|  |
| --- |
| module DC2\_4(  input EN,  input [1:0] DATA,  output reg [3:0] out  );  always@(\*)  if (EN) begin  case(DATA)  2'b00: out <= 4'b0001;  2'b01: out <= 4'b0010;  2'b10: out <= 4'b0100;  2'b11: out <= 4'b1000;  default: out <= out;  endcase |

*Продолжение Листинга 1*

|  |
| --- |
| end  endmodule |

*Листинг 2 – PRCOD4\_2.v*

|  |
| --- |
| module PRCOD4\_2 (  input [3:0] in,  input en,  output reg [1:0] out  );  always @(\*) begin  if(en)  if (in[3])  out = 2'b11;  else if(in[2])  out = 2'b10;  else if(in[1])  out = 2'b01;  else  out = 2'b00;  else  out = 2'b00;  end  endmodule |

*Листинг 3 – RG\_2.v*

|  |
| --- |
| module RG\_2(  input clk, rst, in,  output reg [1:0] out  );  always@(posedge clk or posedge rst)  begin  if(rst)  out <= 2'b00;  else  out <= in;  end  endmodule |

*Листинг 4 – TOP.v*

|  |
| --- |
| module TOP(  input [1:0] D,  input R,  input en,  input clk,  output [1:0] Q  );  wire [3:0] dec\_out;  wire [1:0] reg\_out;  DC2\_4 dc(  .DATA(D),  .out(dec\_out),  .EN(en)  );  PRCOD4\_2 prcod(  .in(dec\_out),  .out(reg\_out), |

*Продолжение Листинга 4*

|  |
| --- |
| .en(en)  );  RG\_2 rg(  .in(reg\_out),  .clk(clk),  .rst(R),  .out(Q)  );  endmodule |

*Листинг 5 – test.v*

|  |
| --- |
| module test;  // input  reg RST, CLK;  reg [1:0] DATA, EN;  //output  wire OUT;  TOP uut(  .D(DATA),  .R(RST),  .Q(OUT),  .en(EN)  );  always begin  #10;  EN = ~EN;  end  initial begin  DATA[0] = 0;  RST = 1;  CLK = 1;  #20;  RST = 0;  #50;  RST = 1;  #40;  DATA[0] = 1;  #40;  $stop;  end  endmodule |

Результат выполнения программы, показано на рисунке 8:

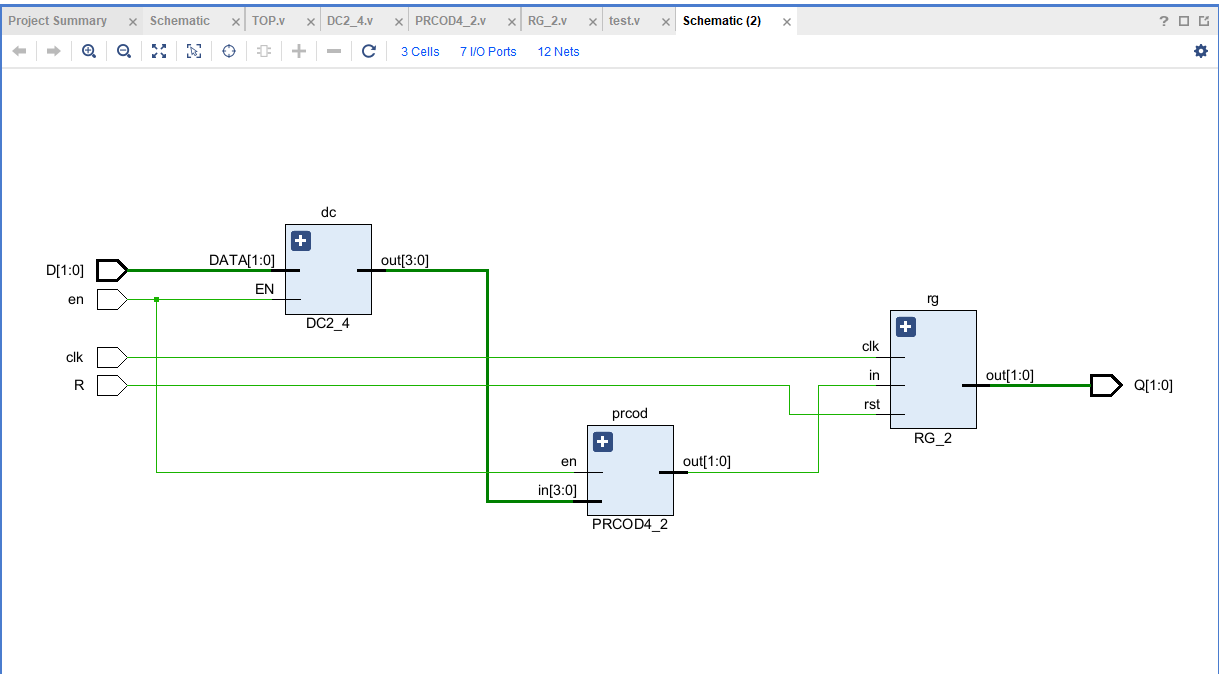


Рисунок 8 – Результат в Schematic Viewer

# **Практическая работа №11**

Задание:

Реализовать модули: делитель частоты (dividor.v) и регистр (RG.v). Выполнить в TOP-модуле, согласно рисунку 9:

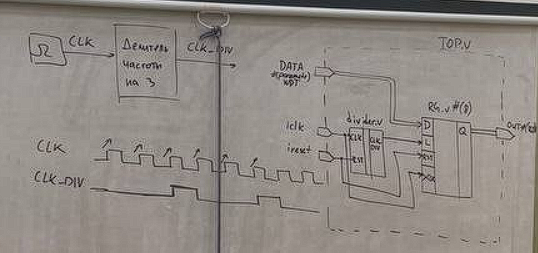


Рисунок 9 – TOP-модуль

Решение:

Данное решение представлено на Листинге – 1, на Листинге – 2, Листинге – 3 и на Листинге – 4.

*Листинг 1 – dividor.v*

|  |
| --- |
| module dividor #(  parameter DIV = 5  )  (  input CLK, RST,  output reg CLK\_DIV  );  reg [$clog2(DIV) - 1:0] counter;  always@(posedge CLK, posedge RST) begin  if(RST) begin  counter <= 0;  CLK\_DIV <= 1'b0;  end  else if(counter == DIV - 1) begin  counter <= 0;  CLK\_DIV <= 1'b1;  end  else begin  counter <= counter + 1;  CLK\_DIV = 1'b0;  end |

*Продолжение Листинга 1*

|  |
| --- |
| end  endmodule |

*Листинг 2 – RG.v*

|  |
| --- |
| module RG#(  parameter WDT\_RG = 8  )(  input L, RST, CLK,  input [WDT\_RG-1:0] D,  output reg [WDT\_RG-1:0] Q  );  always@(posedge CLK, posedge RST) begin  if(RST) begin  Q <= {WDT\_RG{1'b0}};  end  else if(L) begin  Q <= D;  end  end  endmodule |

*Листинг 3 – TOP.v*

|  |
| --- |
| module TOP #(parameter WDT = 8)(  input [WDT - 1:0] DATA,  input iclk,  input irst,  output [WDT - 1:0] out,  wire clk\_div  );  dividor #(.DIV(7)) div(  .CLK(iclk),  .RST(irst),  .CLK\_DIV(clk\_div)  );  RG #(.WDT\_RG(WDT)) rg(  .D(DATA),  .L(clk\_div),  .RST(irst),  .CLK(iclk),  .Q(out)  ); |

*Листинг 4 – test.v*

|  |
| --- |
| module test;  reg [3:0] DATA;  reg CLK;  reg RST;  wire clk\_div;  wire [3:0] OUT;  TOP #(.WDT(4)) dutl(  .iclk(CLK),  .irst(RST),  .out(OUT),  .clk\_div(clk\_div),  .DATA(DATA)  ); |

*Продолжение Листинга 4*

|  |
| --- |
| always begin  #10;  CLK = ~CLK;  end  initial begin  CLK = 0;  RST = 0;  #50;  RST = 1;  #100;  RST = 0;  DATA = 1;  #50;  RST = 1;  #50;  RST = 0;  DATA = 3;  #300;  $stop;  end  endmodule |

Результат выполнения программы, показано на рисунке 10:

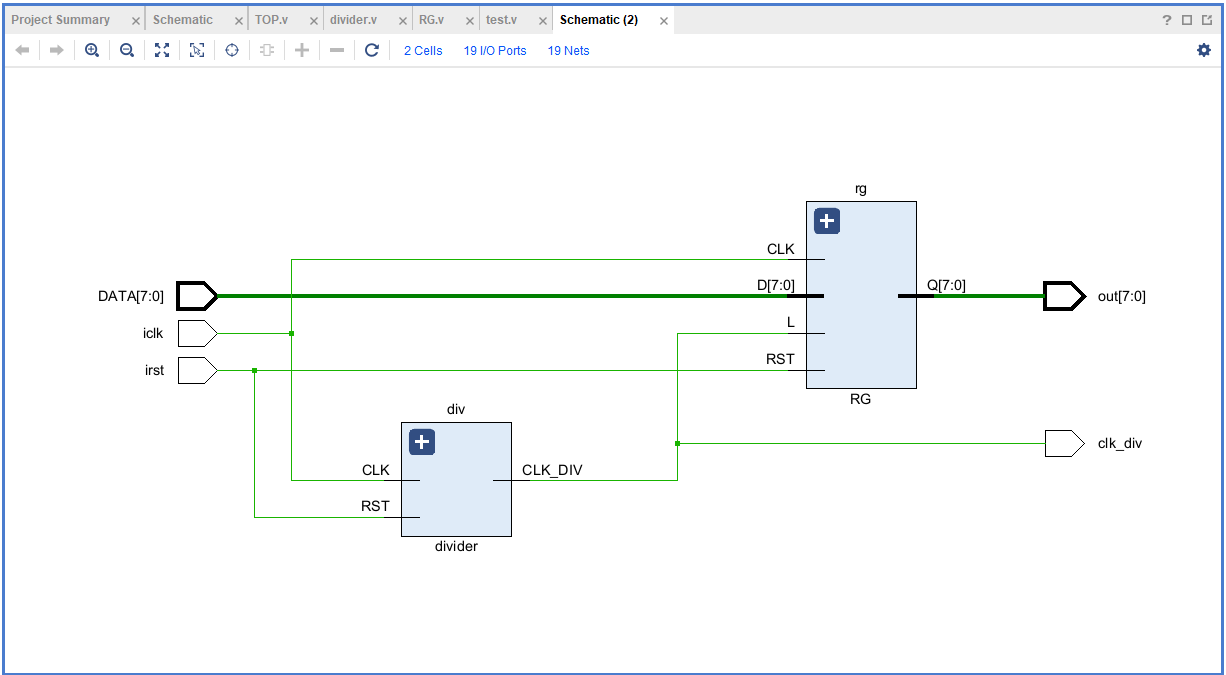


Рисунок 10 – Результат лабораторной работы №3

# **СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ**

1. Федеральный закон №127 от 23 августа 1996 г. «О науке и государственной научно-технической политике» (ред. от 23.05.2016) ― URL: http://www.consultant.ru/document/cons\_doc\_LAW\_11507/ (Дата обращения: 07.02.2019)
2. Громов Г. Р. Очерки информационной технологии. — М.: ИнфоАрт, 2015. — 336 с.
3. Информатика: Учебник / Под ред. проф. Н. В. Макаровой. — М.: Финансы и статистика, 2015. — 768 с.
4. Качала В. В. Предварительное обследование при реорганизации управления предприятием // Третья Российская научно-практическая конференция «Реинжиниринг бизнес-процессов на основе современных информационных технологий». — М.: МЭСИ, 2012. — С. 248–253.
5. Надарая Э. А. Об оценке регрессии // Теория вероятностей и ее применения. — 2010. — Т. 9. — Вып. 1. — С. 157–159.
6. Пур А. Накопители XXI века // PC Magazine. — 2013. — № 4. — С.138146.
7. Фурсов К.С. Анализ новейших международных рекомендаций в области статистического измерения исследований и разработок (Руководство Фраскати) и возможность их адаптации в отечественной статистике ― URL: http://www.gks.ru/free\_doc/new\_site/rosstat/nms/prez2\_1503.pdf (Дата обращения: 07.02.2019)
8. Billings S. A., Fadzil M. B., Sulley J., Johnson P. M. Identification of a non-linear difference equation model of an industrial diesel generator // Mechanical Systems and Signal Processing. — 2015. – Vol. 2. — N 1. — P. 59–76.