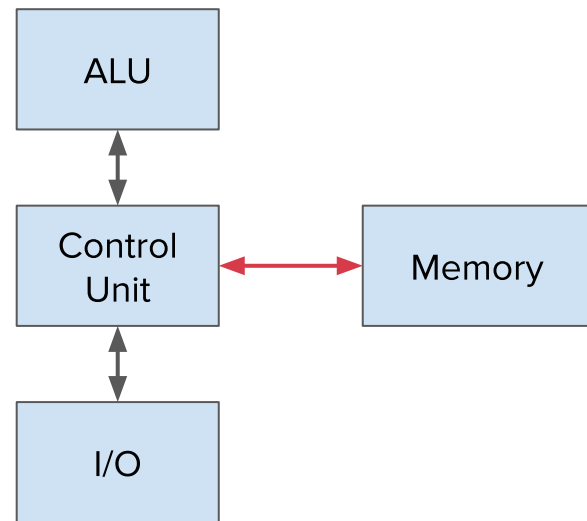
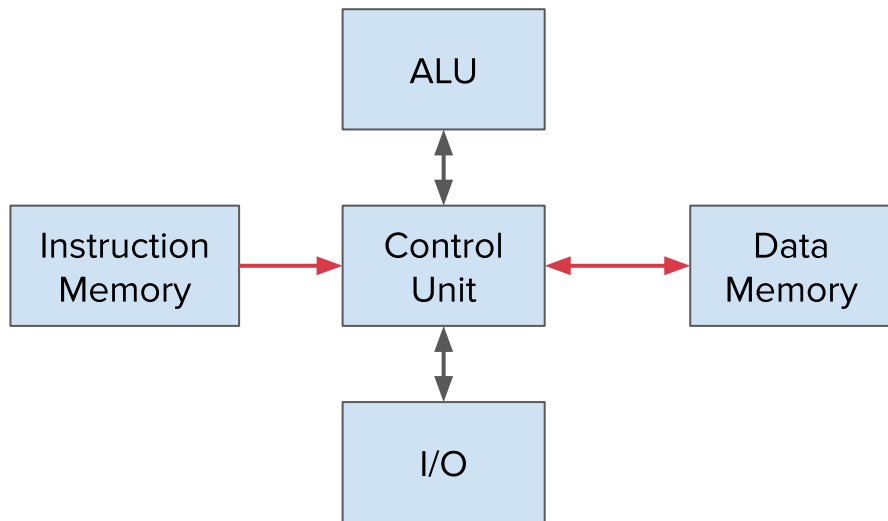


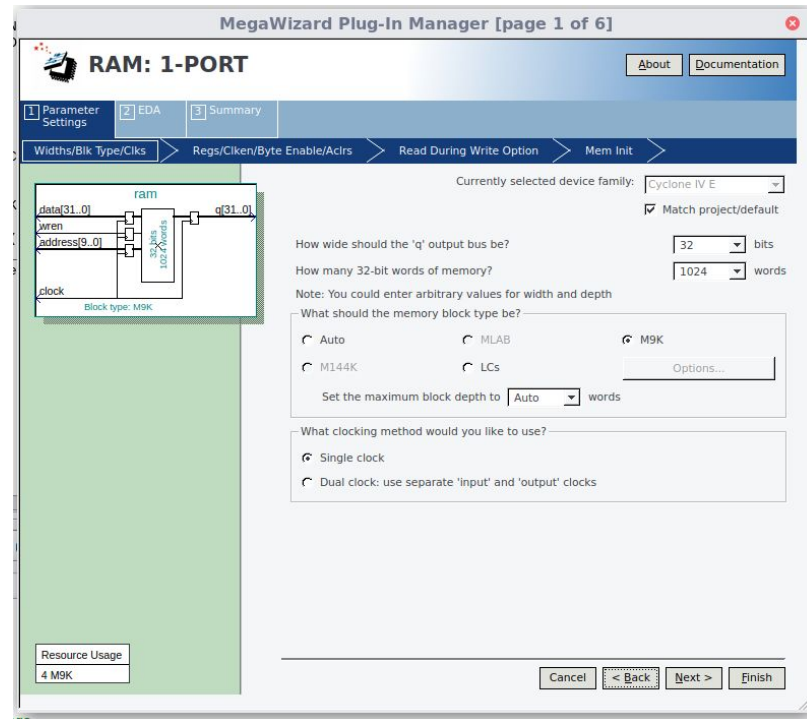
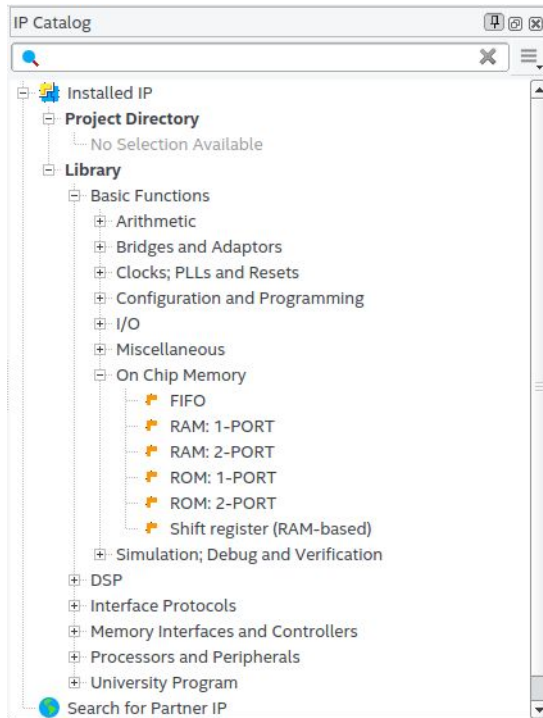
RAM. Память данных. Стек

# Память данных

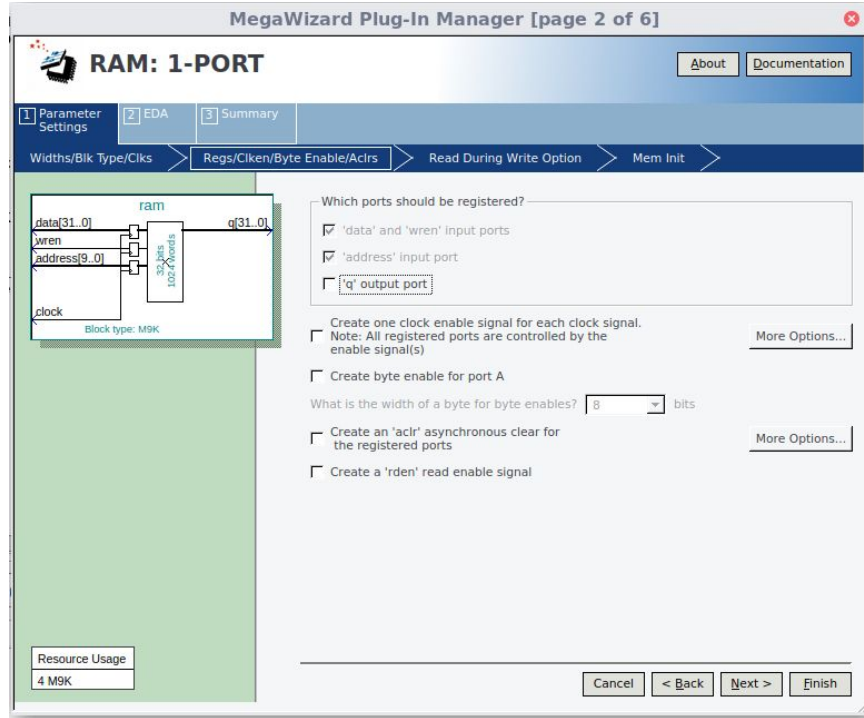
- **Архитектура фон Неймана** — данные и команды хранятся совместно
- **Гарвардская архитектура** — данные и команды хранятся отдельно, каналы данных и инструкций тоже разделены



# RAM: 1-PORT



# RAM: 1-PORT

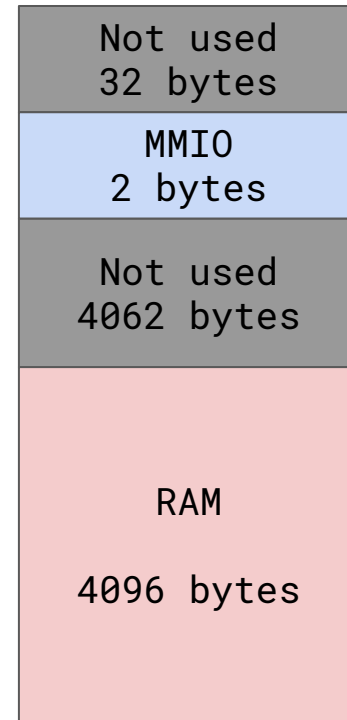


```
module ram(  
    input    [9:0]address,  
    input    clock,  
    input    [31:0]data,  
    input    wren,  
    output   [31:0]q  
);
```

# mem\_ctrl.v

```
always @(*) begin
    q = 32'b0;
    mmio_data = 16'b0;
    mmio_we = 1'b0;
    ram_data = 32'b0;
    ram_addr = 10'b0;
    ram_we = 1'b0;

    casez (addr)
        32'h20: begin // MMIO
            mmio_data = data[15:0];
            mmio_we = we;
        end
        32'b1_????_????_????: begin // RAM
            ram_addr = addr[11:2];
            ram_data = data;
            ram_we = we;
            q = ram_q;
        end
    endcase
end
```



# control.v

```
7'b0000011: begin // LW
    imm12 = instr[31:20];
    rf_we = (funct3 == 3'b010);
    has_imm = 1'b1;
    is_load = 1'b1;
end
7'b0100011: begin // SW
    imm12 = {instr[31:25], instr[11:7]};
    has_imm = 1'b1;
    mem_we = (funct3 == 3'b010);
end
```

simm[11:0]		rs1	000	rd	0000011	LB rd, offset(rs1)
simm[11:0]		rs1	001	rd	0000011	LH rd, offset(rs1)
simm[11:0]		rs1	010	rd	0000011	LW rd, offset(rs1)
simm[11:0]		rs1	100	rd	0000011	LBU rd, offset(rs1)
simm[11:0]		rs1	101	rd	0000011	LHU rd, offset(rs1)
simm[11:5]	rs2	rs1	000	simm[4:0]	0100011	SB rs2, offset(rs1)
simm[11:5]	rs2	rs1	001	simm[4:0]	0100011	SH rs2, offset(rs1)
simm[11:5]	rs2	rs1	010	simm[4:0]	0100011	SW rs2, offset(rs1)

# core.v

```
wire [31:0]pc_next = ((pc == LAST_PC) || (is_load && ~load_delay)) ? pc : pc_target;
```

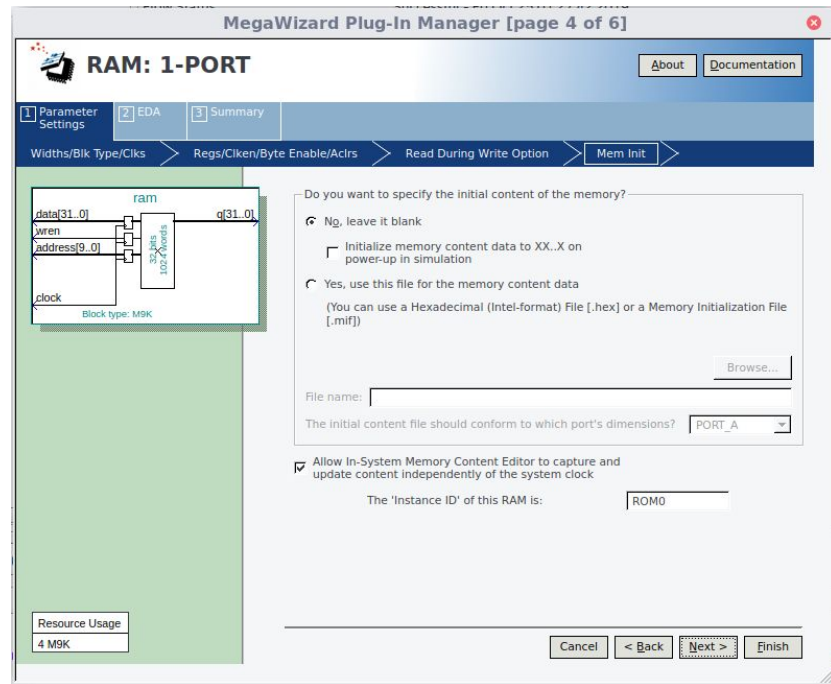
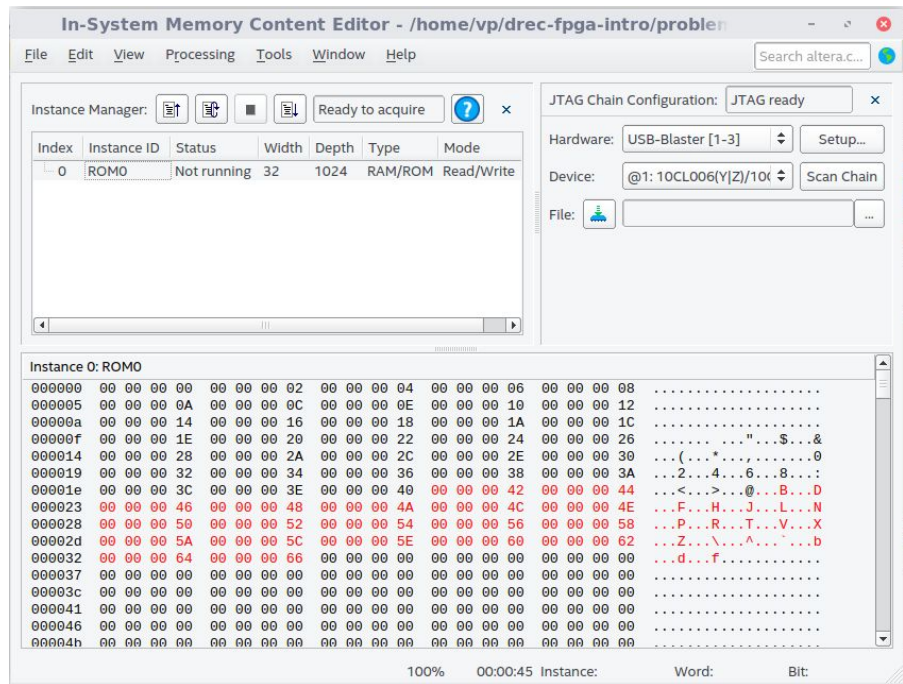
```
reg load_delay = 1'b0;
```

```
always @(posedge clk) begin
    pc <= pc_next;
    load_delay <= is_load && ~load_delay;
end
```

```
.....
```

```
wire [31:0]rf_wdata = is_load ? mem_q : alu_result;
```

# In-System Memory Content Editor





# Пример функции

```
typedef unsigned int uint32_t;  
typedef unsigned char uint8_t;
```

```
uint32_t mean4(uint32_t x0, uint32_t x1, uint32_t x2, uint32_t x3) {  
    uint32_t mean;  
  
    mean = (x0 + x1 + x2 + x3) >> 2;  
  
    return mean;  
}
```

```
$ riscv64-linux-gnu-gcc -c mean4.c -march=rv32i -mabi=ilp32 -nostdlib -o mean4.o  
$ riscv64-linux-gnu-objdump -d mean4.o
```

# Стековый фрейм

00000000 <mean4>:

0:	fd010113	addi	sp, sp, -48
4:	02812623	sw	s0, 44(sp)
8:	03010413	addi	s0, sp, 48
c:	fca42e23	sw	a0, -36(s0)
10:	fc42c23	sw	a1, -40(s0)
14:	fcc42a23	sw	a2, -44(s0)
18:	fcd42823	sw	a3, -48(s0)
1c:	fdc42703	lw	a4, -36(s0)
20:	fd842783	lw	a5, -40(s0)
24:	00f70733	add	a4, a4, a5
28:	fd442783	lw	a5, -44(s0)
2c:	00f70733	add	a4, a4, a5
30:	fd042783	lw	a5, -48(s0)
34:	00f707b3	add	a5, a4, a5
38:	0027d793	srl	a5, a5, 0x2
3c:	fef42623	sw	a5, -20(s0)
40:	fec42783	lw	a5, -20(s0)
44:	00078513	mv	a0, a5
48:	02c12403	lw	s0, 44(sp)
4c:	03010113	addi	sp, sp, 48
50:	00008067	ret	



# GitHub

[github.com/viktor-prutyanov/drec-fpga-intro](https://github.com/viktor-prutyanov/drec-fpga-intro)