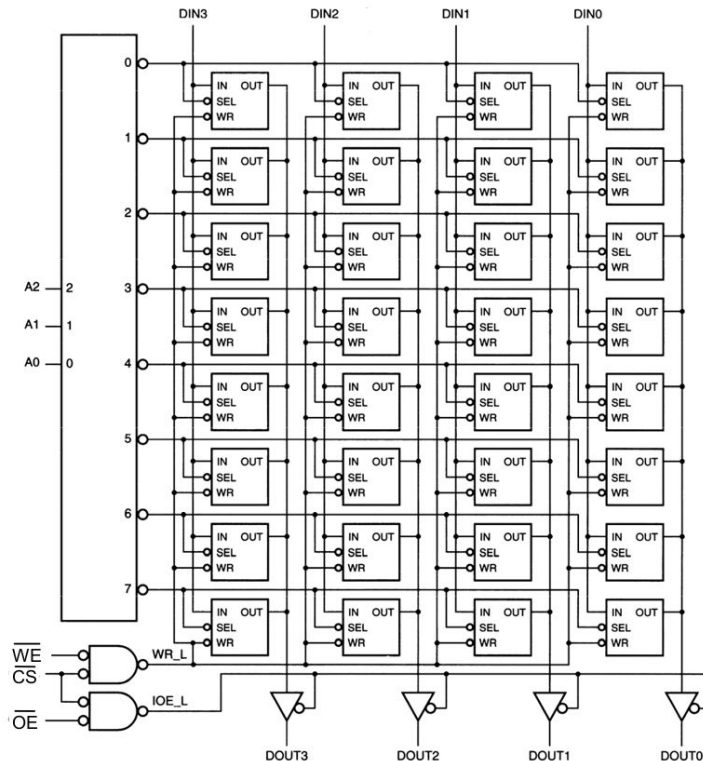


RAM. RAM в FPGA.

Память команд

Память

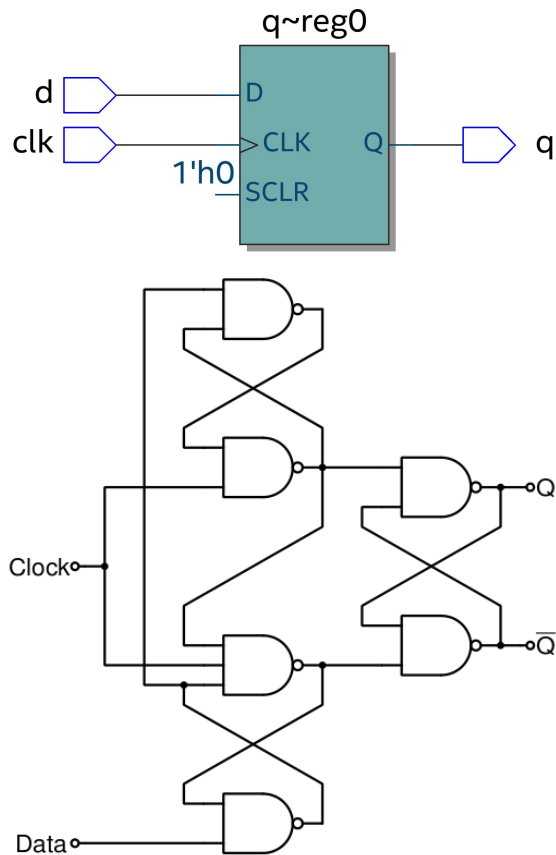
- RAM - Random Access Memory
 - ROM - Read Only Memory
 - Пример: микросхема с BIOS
 - Триггеры
 - Динамическое ОЗУ (DRAM)
 - Статическое ОЗУ (SRAM)
-
- Адресные входы
 - Входы данных
 - Выходы данных



Матрица SRAM 8x4

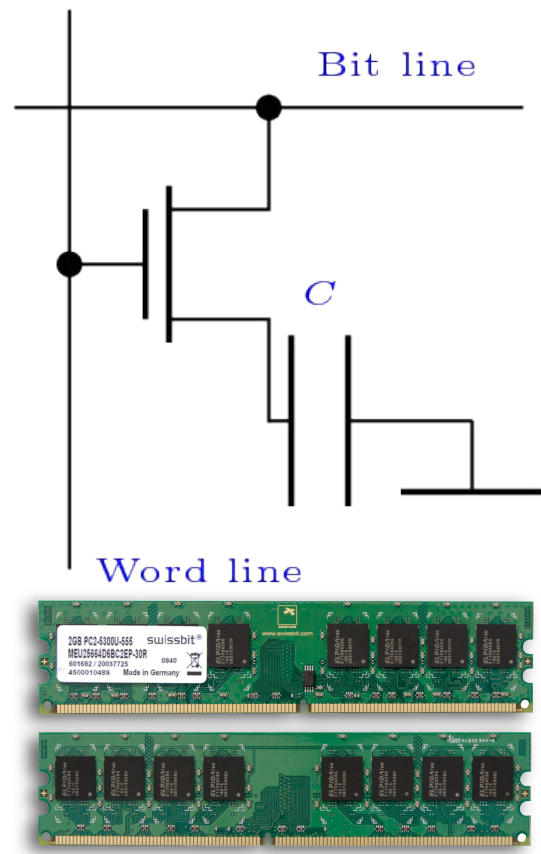
Триггеры

- Триггеры
 - ~20 транзисторов
 - Маленькая задержка
 - Пример: регистровый файл
- Динамическое ОЗУ (DRAM)
- Статическое ОЗУ (SRAM)



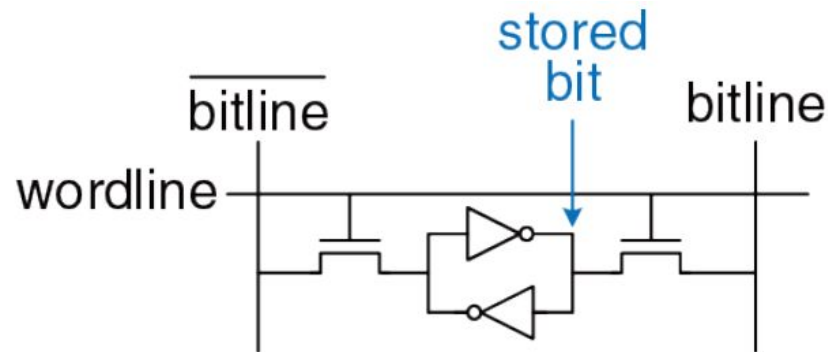
DRAM

- Триггеры
- Динамическое ОЗУ (DRAM)
 - Битовым значениям соответствует наличие или отсутствие заряда конденсатора, управляемого полевым транзистором.
 - Из-за саморазряда ячейки такой памяти нужно регенерировать.
 - 1 транзистор на ячейку
 - Большая задержка
- Статическое ОЗУ (SRAM)



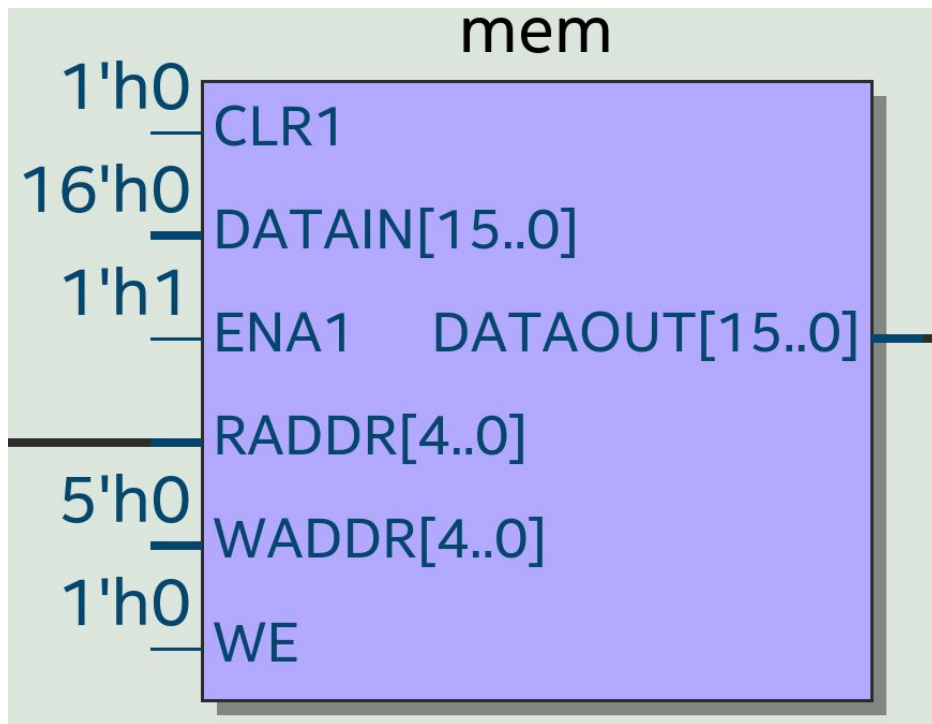
SRAM

- Триггеры
- Динамическое ОЗУ (DRAM)
- Статическое ОЗУ (SRAM)
 - Данные хранятся в бистабильной схеме
 - 6 транзисторов на ячейку
 - Средняя задержка



M9K SRAM в FPGA Altera Cyclone IV

- Всего 276480 бит в EP4CE6E22C8
- Блоки по 9216 бит
- Работа в режиме RAM и ROM
- Инициализация из файла в момент прошивки FPGA



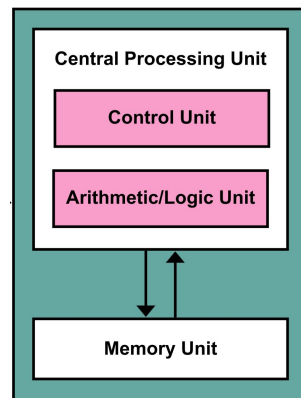
Память команд и память данных

- Память команд
- Память данных

```
int main() {  
    *(char *)0x1080 = 0x30;  
}
```

```
b8 80 10 00 00    mov eax, 0x1000  
c6 00 30          mov BYTE PTR [eax], 0x30
```

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



rom.v

```
module rom #(parameter LENGTH = 8,  
             parameter WIDTH = 32)(  
    input  [31:0]addr,  
  
    output [WIDTH - 1:0]q  
);  
  
reg [WIDTH - 1:0]mem[LENGTH - 1:0];  
  
assign q = mem[addr];  
  
initial begin  
    $readmemh("program.txt", mem);  
end  
  
endmodule
```

```
module rom_fetcher(  
    input clk,  
  
    output [15:0]q  
);  
  
reg [7:0]pc = 0;  
wire [7:0]pc_next = pc + 1;  
  
rom #(.WIDTH(16)) rom(.addr(pc), .q(q));  
  
always @(posedge clk) begin  
    pc <= pc_next;  
end  
  
endmodule
```


Результат

```
$ cat program.txt  
BEEF  
1001  
1234  
5678  
9ABC  
DEF0  
6000  
7000
```

