Продолжение временных параметоров

Параметры синхросигнала

- Частота (обратная величина периоду)
- Скважность (обратная величина коэффициенту заполнения)
- Джиттер
- Скорость нарастания и спада фронта (transition)

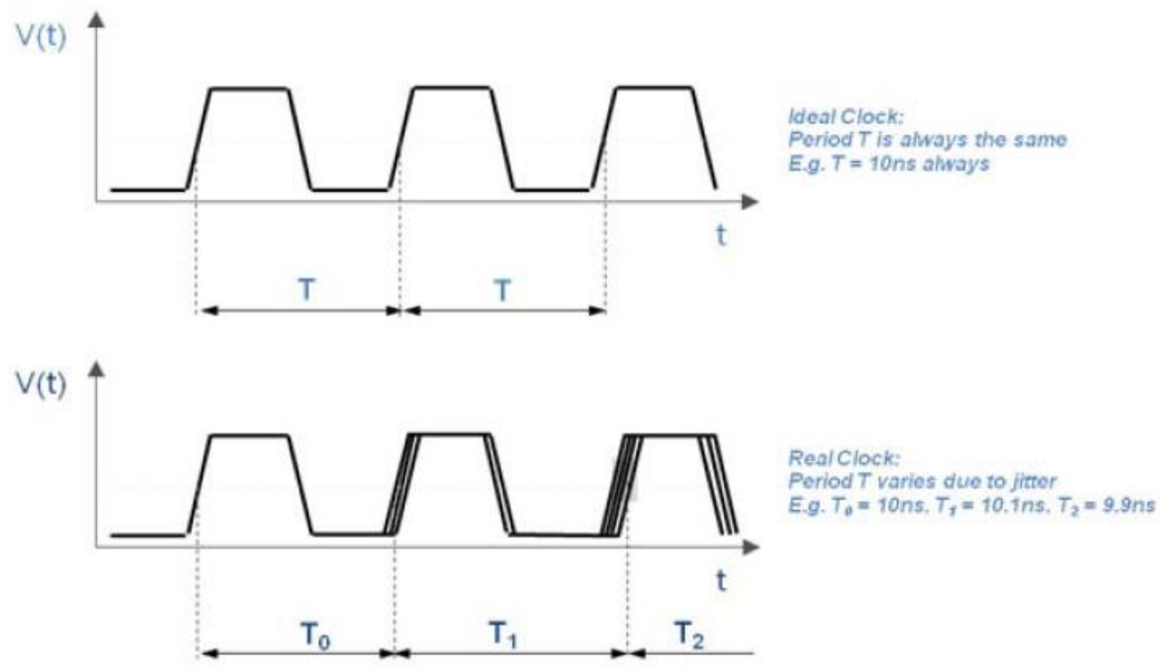
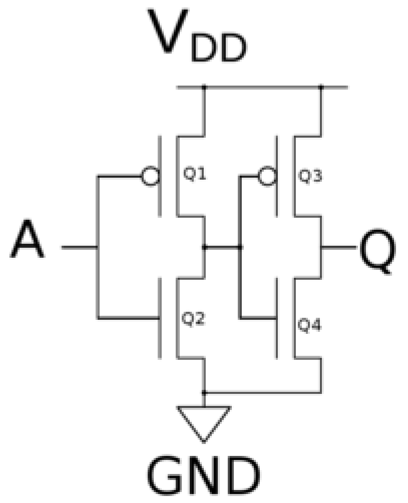

V(t)
t
T
T
Ideal Clock:
Period T is always the same
E.g. T = 10ns always
V(t)
t
T_0
T_1
T_2
Real Clock:
Period T varies due to jitter
E.g. T_0 = 10ns, T_1 = 10.1ns, T_2 = 9.9ns

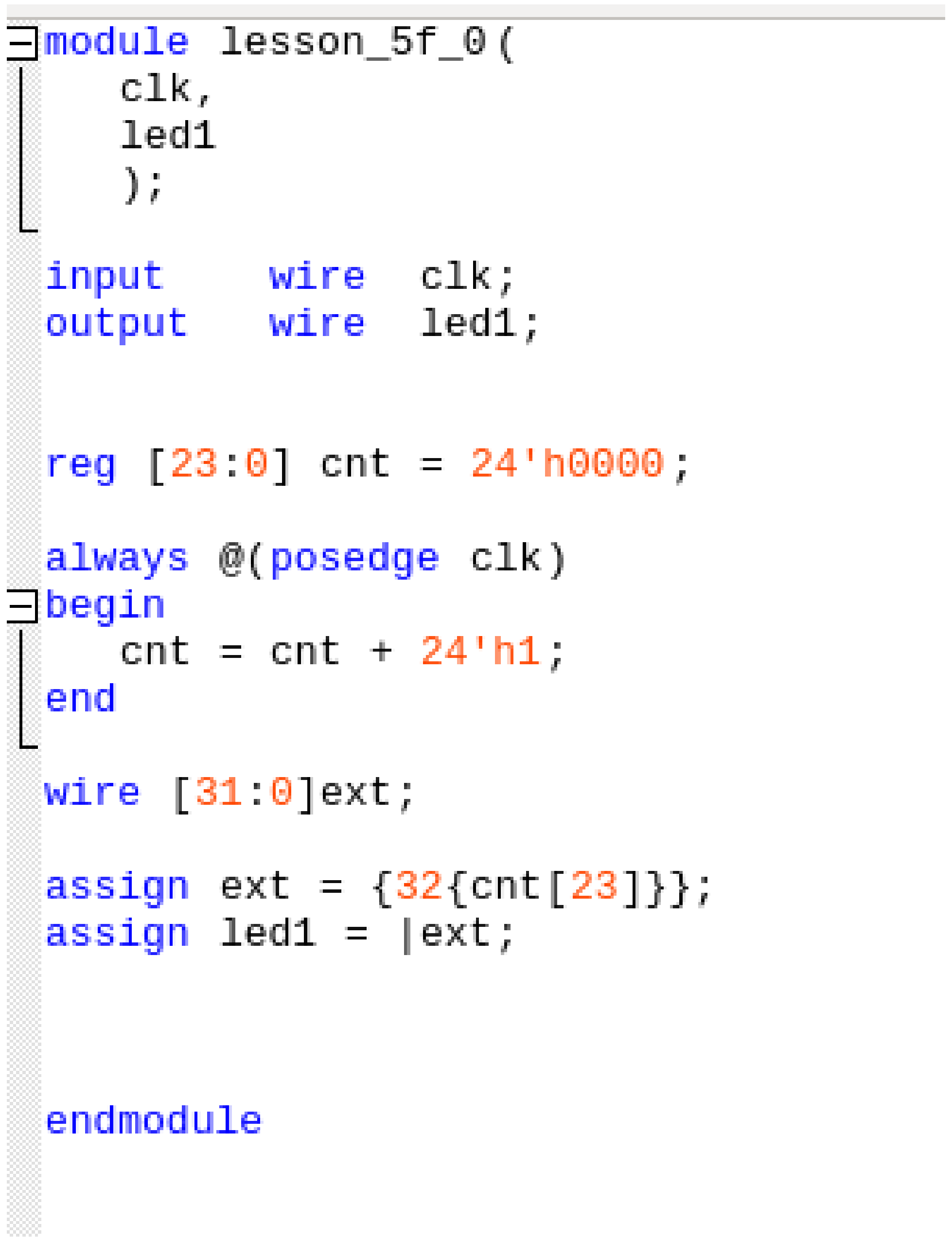
Figure 1. Jitter in the Time Domain

Причины возникновения джиттер

- Тепловой шум
- RC-цепь – делать линии шире
- Большая выходная нагрузка элемента в пути синхросигнала – уменьшать нагрузку.
- Низкая скорость нарастания фронта синхросигнала – увеличить скорость нарастания фронта

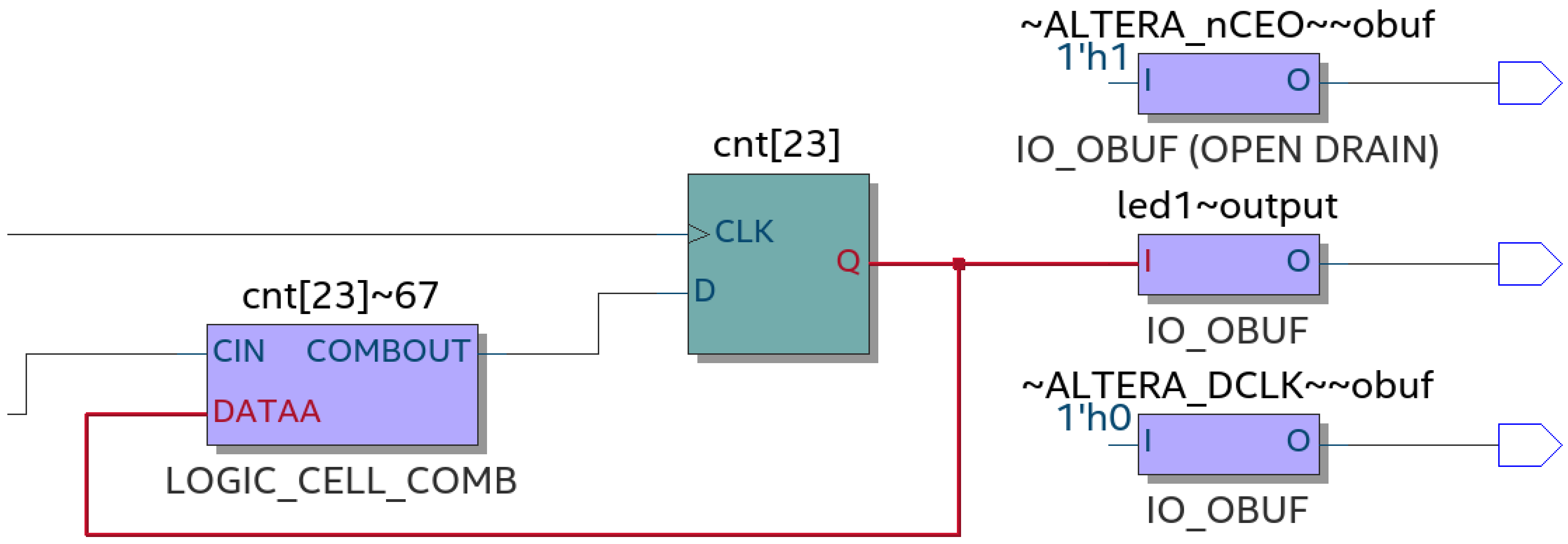


MAX FANOUT

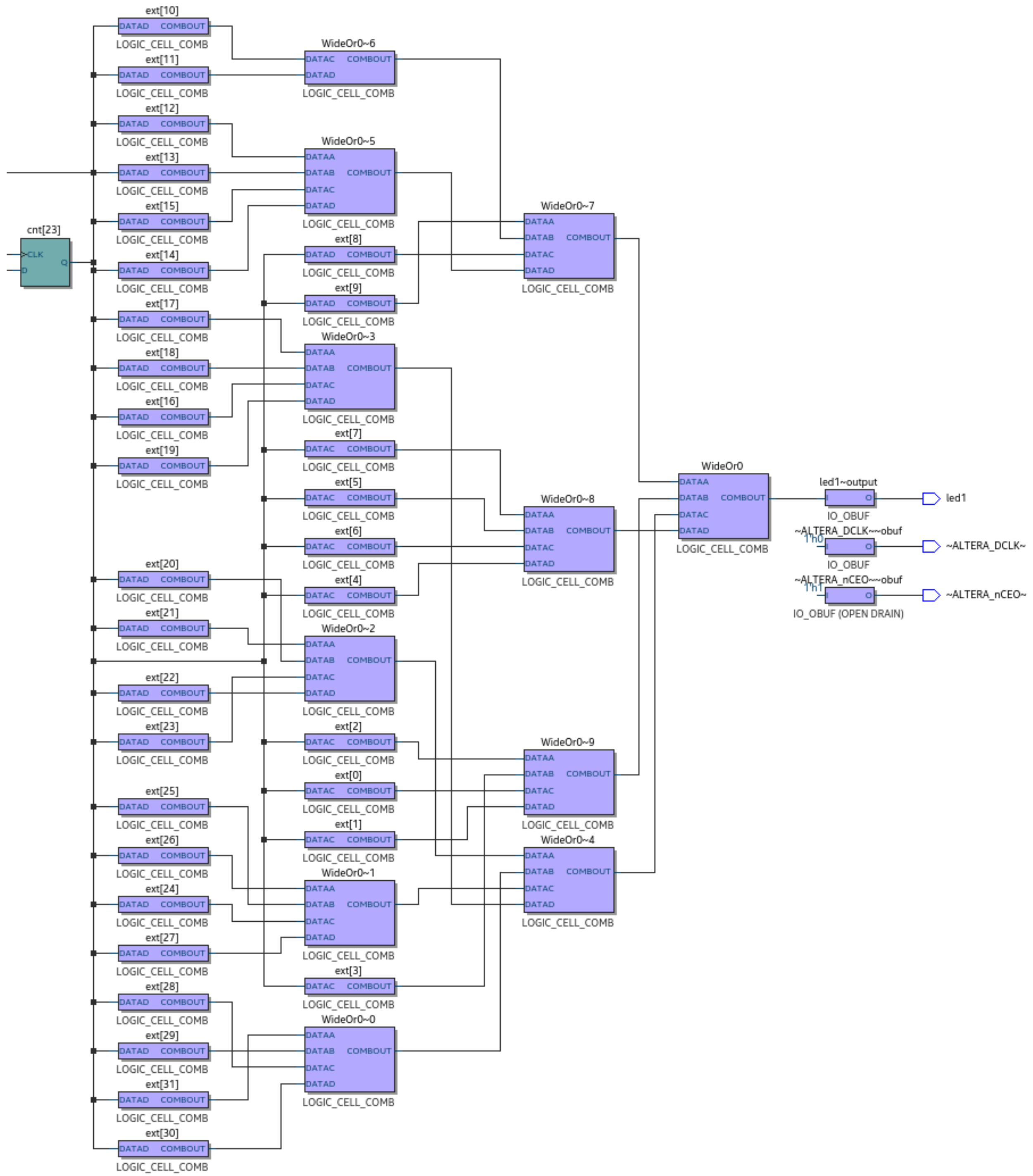


```
module lesson_5f_0(
    clk,
    led1
    );

input     wire  clk;
output    wire  led1;


reg [23:0] cnt = 24'h0000;

always @(posedge clk)
begin
    cnt = cnt + 24'h1;
end

wire [31:0]ext;

assign ext = {32{cnt[23]}};
assign led1 = |ext;



endmodule
```

MAX FANOUT

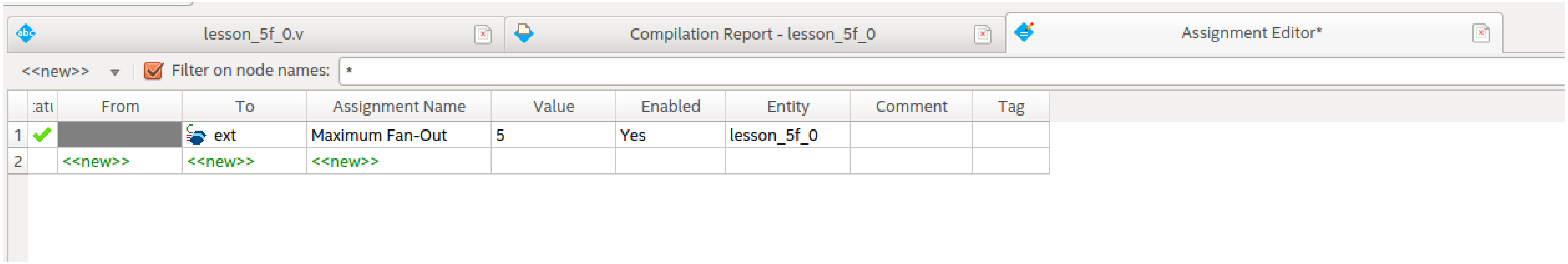

~ALTERA_nCEO~~obuf
1'h1
I
O
IO_OBUF (OPEN DRAIN)
cnt[23]
CLK
Q
D
led1~output
I
O
IO_OBUF
cnt[23]~67
CIN
COMBOUT
DATAA
LOGIC_CELL_COMB
~ALTERA_DCLK~~obuf
1'h0
I
O
IO_OBUF

MAX FANOUT



cnt[23]
ext[0]–ext[31]
WideOr0~0 … WideOr0~9
WideOr0
LOGIC_CELL_COMB
led1~output
IO_OBUF
led1
~ALTERA_DCLK~~obuf
~ALTERA_DCLK~
~ALTERA_nCEO~~obuf
IO_OBUF (OPEN DRAIN)
~ALTERA_nCEO~

MAX FANOUT

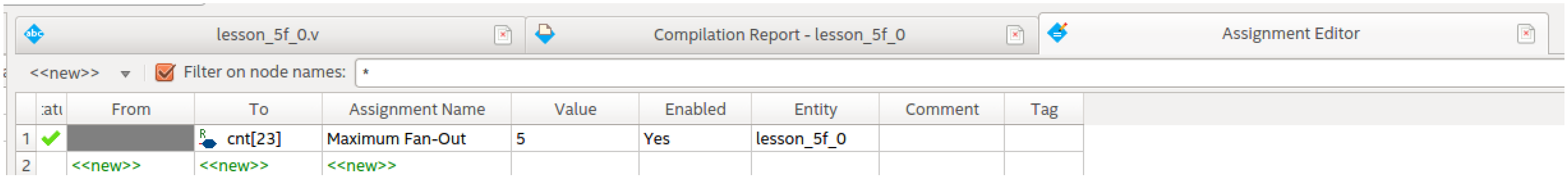


lesson_5f_0.v | Compilation Report - lesson_5f_0 | Assignment Editor*

<<new>> Filter on node names: *

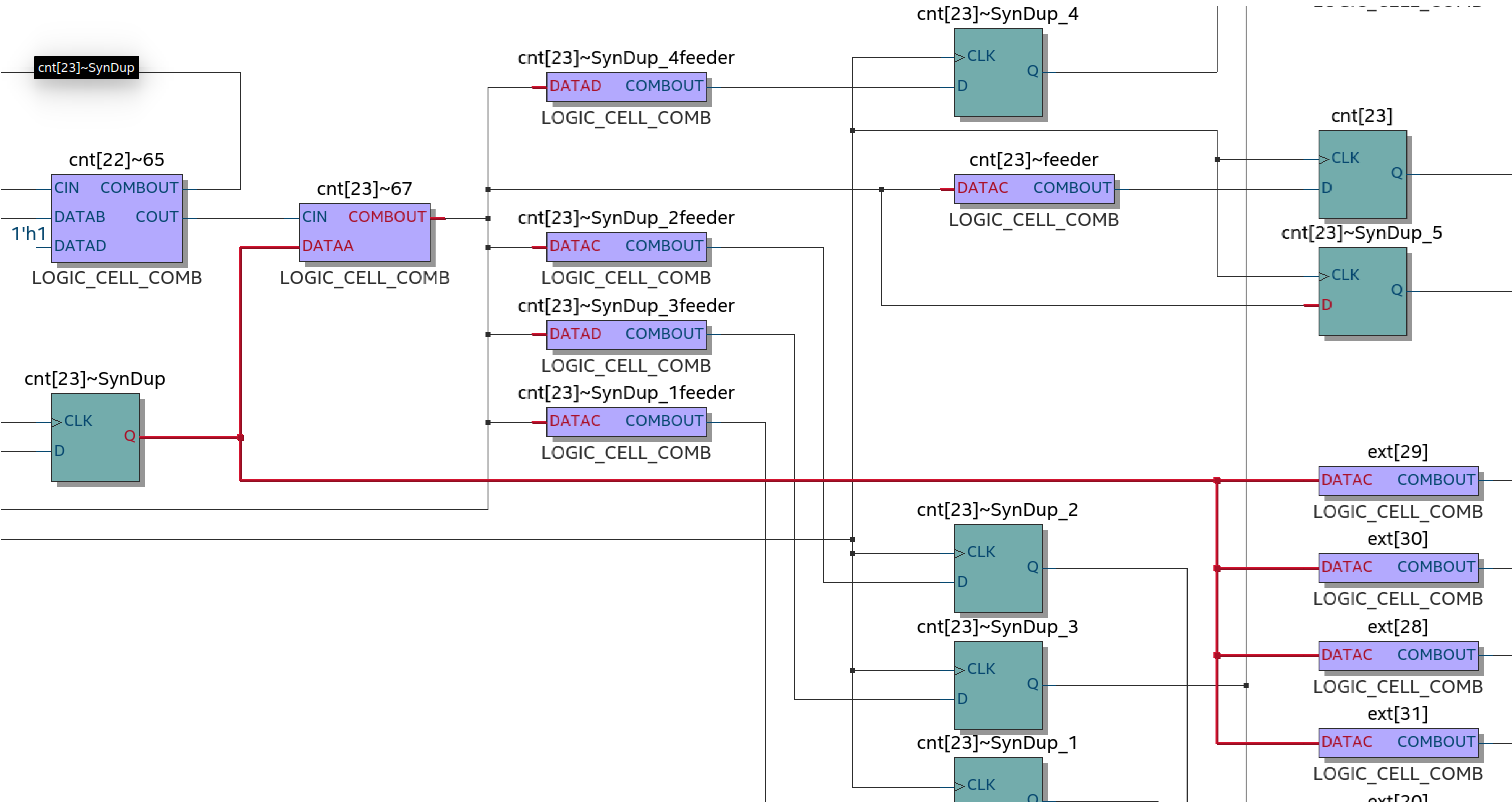
	:atu	From	To	Assignment Name	Value	Enabled	Entity	Comment	Tag
1	✓		ext	Maximum Fan-Out	5	Yes	lesson_5f_0		
2		<<new>>	<<new>>	<<new>>					

MAX FANOUT

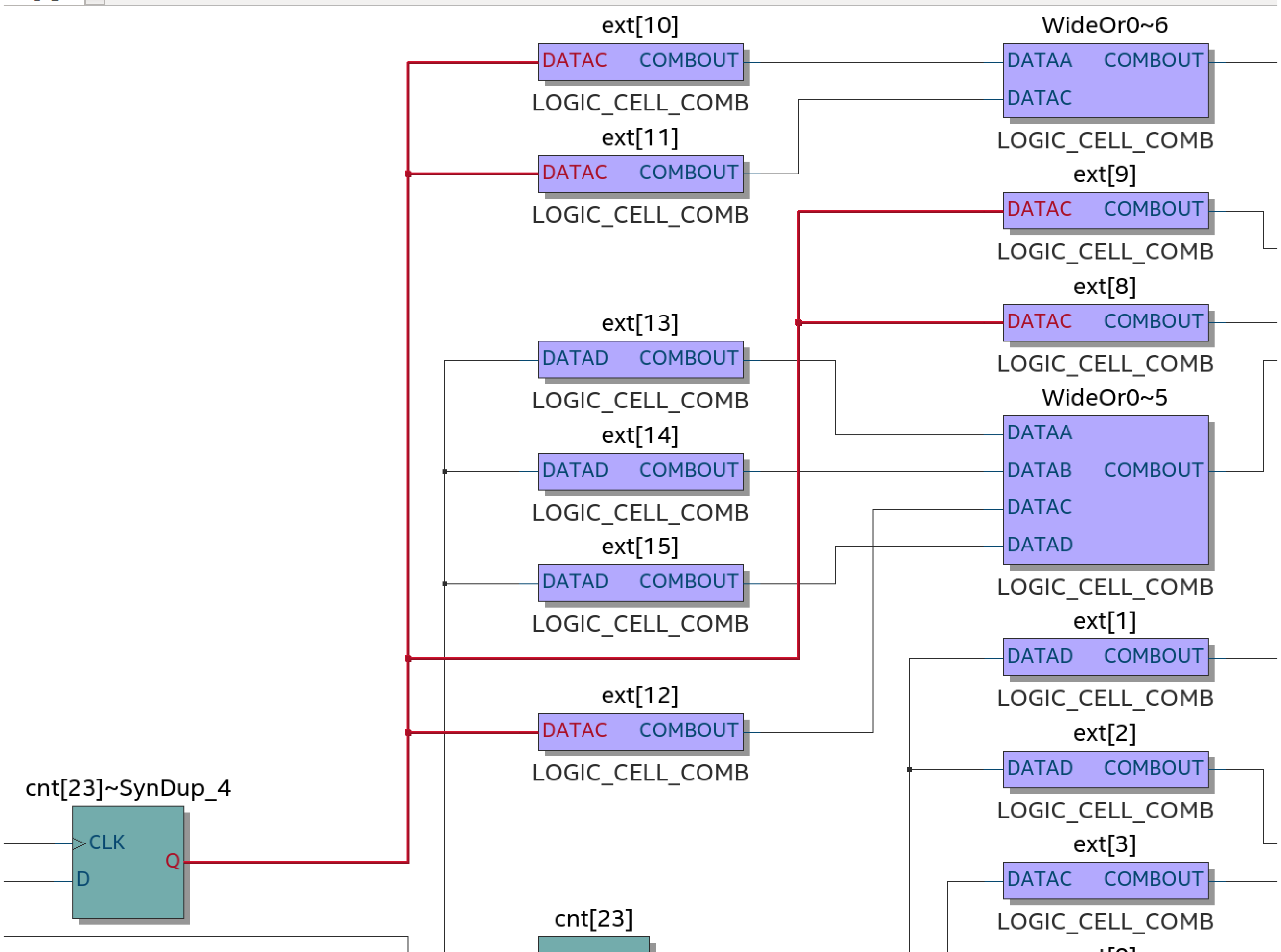


Filter on node names: *

	:atu	From	To	Assignment Name	Value	Enabled	Entity	Comment	Tag
1	✓		cnt[23]	Maximum Fan-Out	5	Yes	lesson_5f_0		
2		<<new>>	<<new>>	<<new>>					



cnt[23]~SynDup
cnt[23]~SynDup_4
cnt[23]~SynDup_4feeder
DATAD COMBOUT
LOGIC_CELL_COMB
cnt[23]
cnt[22]~65
CIN COMBOUT
DATAB COUT
1'h1
DATAD
LOGIC_CELL_COMB
cnt[23]~67
CIN COMBOUT
DATAA
LOGIC_CELL_COMB
cnt[23]~feeder
DATAC COMBOUT
LOGIC_CELL_COMB
cnt[23]~SynDup_2feeder
DATAC COMBOUT
LOGIC_CELL_COMB
cnt[23]~SynDup_5
cnt[23]~SynDup_3feeder
DATAD COMBOUT
LOGIC_CELL_COMB
cnt[23]~SynDup
CLK Q D
cnt[23]~SynDup_1feeder
DATAC COMBOUT
LOGIC_CELL_COMB
ext[29]
DATAC COMBOUT
LOGIC_CELL_COMB
cnt[23]~SynDup_2
ext[30]
DATAC COMBOUT
LOGIC_CELL_COMB
cnt[23]~SynDup_3
ext[28]
DATAC COMBOUT
LOGIC_CELL_COMB
ext[31]
DATAC COMBOUT
LOGIC_CELL_COMB
cnt[23]~SynDup_1

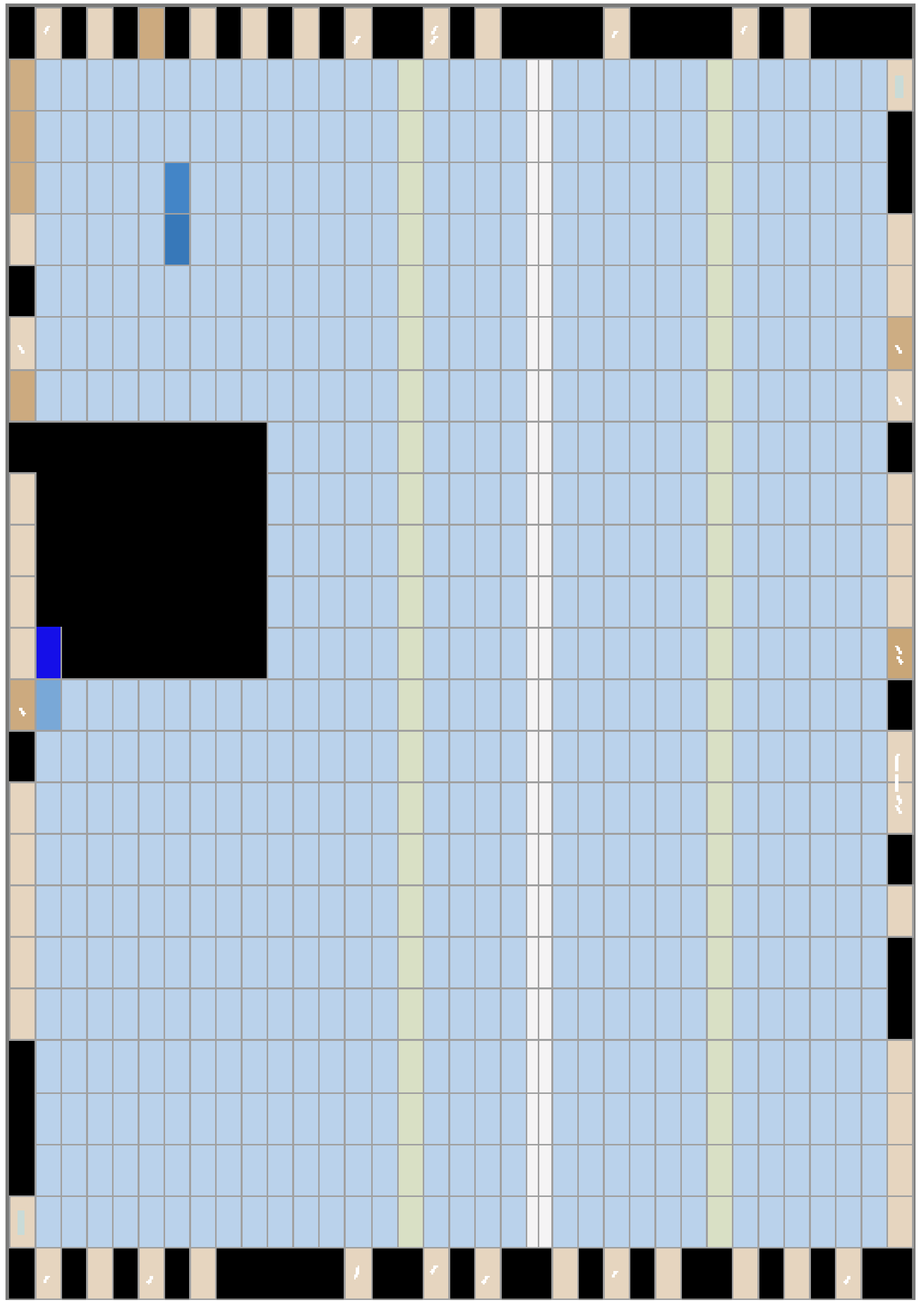


ext[10]
DATAC
COMBOUT
LOGIC_CELL_COMB
WideOr0~6
DATAA
COMBOUT
DATAC
LOGIC_CELL_COMB
ext[11]
DATAC
COMBOUT
LOGIC_CELL_COMB
ext[9]
DATAC
COMBOUT
LOGIC_CELL_COMB
ext[8]
DATAC
COMBOUT
LOGIC_CELL_COMB
ext[13]
DATAD
COMBOUT
LOGIC_CELL_COMB
WideOr0~5
DATAA
DATAB
COMBOUT
DATAC
DATAD
LOGIC_CELL_COMB
ext[14]
DATAD
COMBOUT
LOGIC_CELL_COMB
ext[15]
DATAD
COMBOUT
LOGIC_CELL_COMB
ext[1]
DATAD
COMBOUT
LOGIC_CELL_COMB
ext[12]
DATAC
COMBOUT
LOGIC_CELL_COMB
ext[2]
DATAD
COMBOUT
LOGIC_CELL_COMB
cnt[23]~SynDup_4
CLK
Q
D
ext[3]
DATAC
COMBOUT
LOGIC_CELL_COMB
cnt[23]

Память

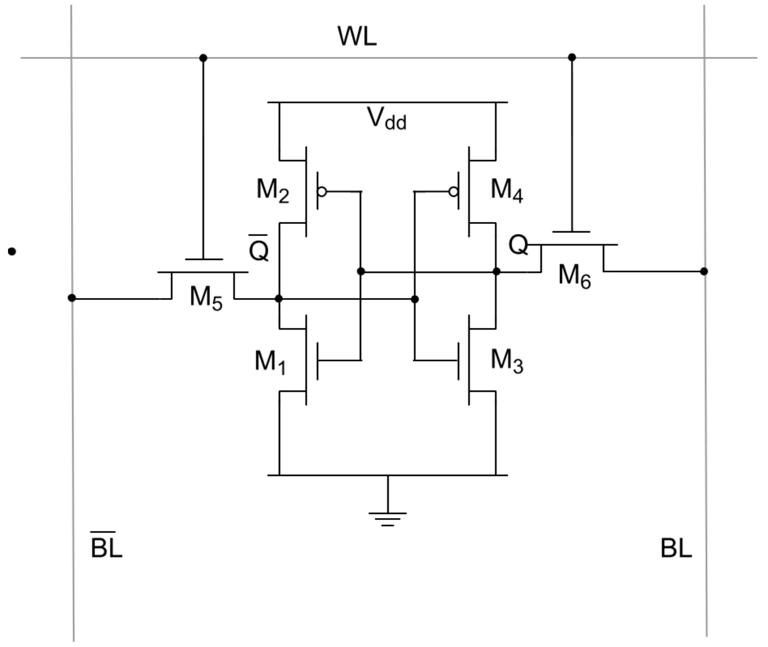
Что будем разбирать

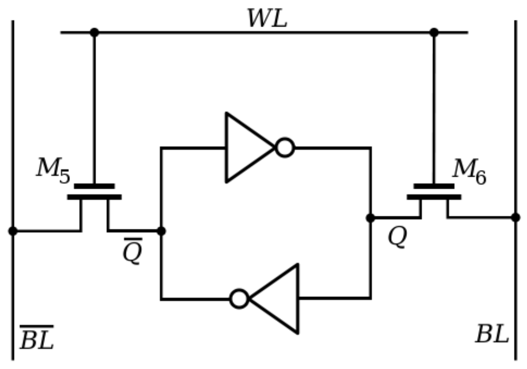
- Flash для прошивки
- M9K (режимы работы ROM, RAM, SR, FIFO)
- Внешняя RAM



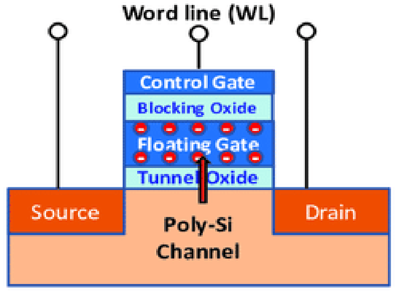
Прошивка ПЛИС

- SOF file – конфигурирует SRAM (полупроводниковая память, энергозависимая, регенерация не нужна)
- JIC file – конфигурирует flash (транзистор с плавающим затвором, энергонезависимая, регенерация не нужна)

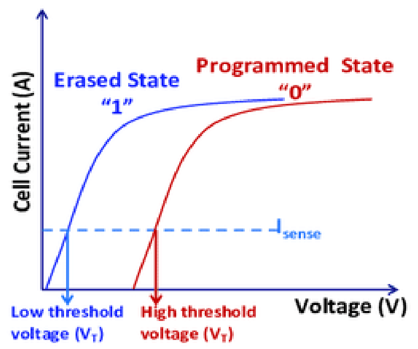




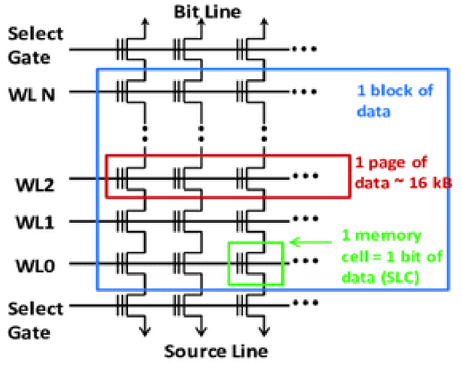
(a) A floating gate flash memory cell



(b) I-V characteristics of memory cell

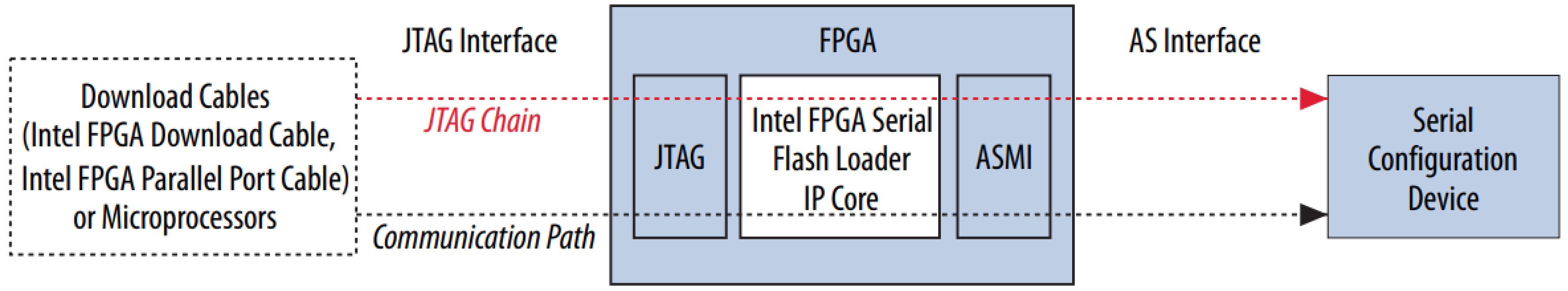


(c) NAND Flash memory array



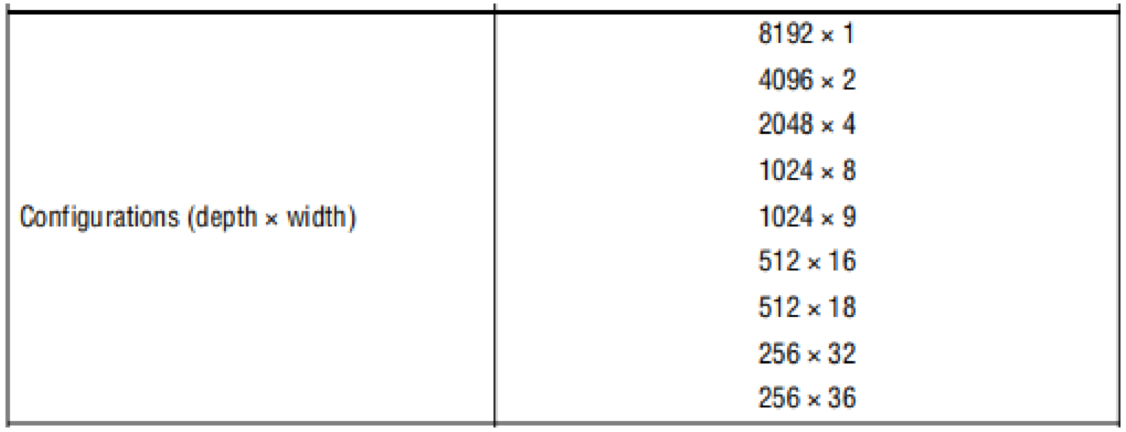
Конфигурация flash

- Implement altera serial flash loader ip
- Convert configuration file to .jic format
- Configure JTAG chain
- Configure memory


JTAG Interface
FPGA
AS Interface
Download Cables (Intel FPGA Download Cable, Intel FPGA Parallel Port Cable) or Microprocessors
JTAG Chain
JTAG
Intel FPGA Serial Flash Loader IP Core
ASMI
Serial Configuration Device
Communication Path

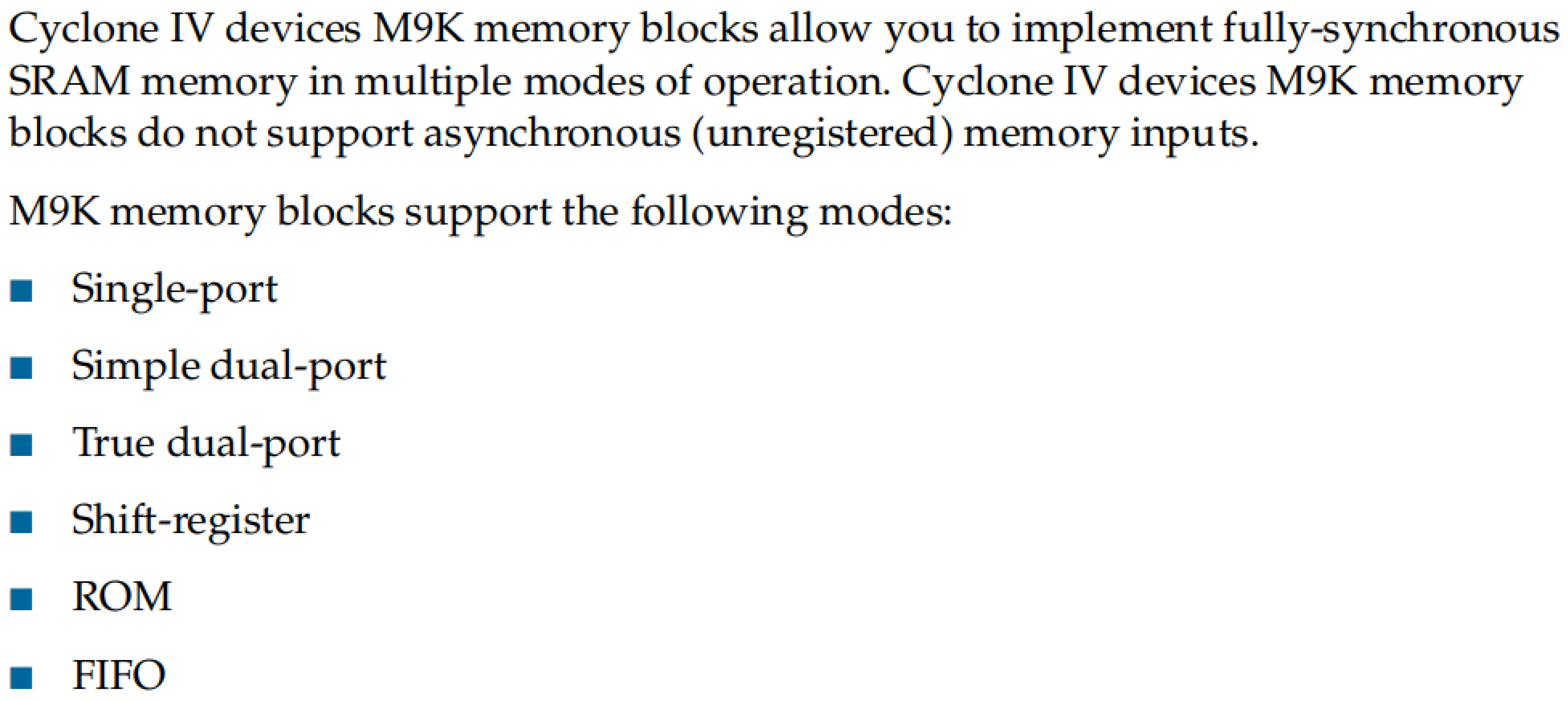
M9K

- SOF file – конфигурирует SRAM (полупроводниковая память, энергозависимая, регенерация не нужна)
- JIC file – конфигурирует flash (транзистор с плавающим затвором, энергонезависимая, регенерация не нужна)



Configurations (depth × width)	8192 × 1 4096 × 2 2048 × 4 1024 × 8 1024 × 9 512 × 16 512 × 18 256 × 32 256 × 36

M9K modes



Cyclone IV devices M9K memory blocks allow you to implement fully-synchronous SRAM memory in multiple modes of operation. Cyclone IV devices M9K memory blocks do not support asynchronous (unregistered) memory inputs.

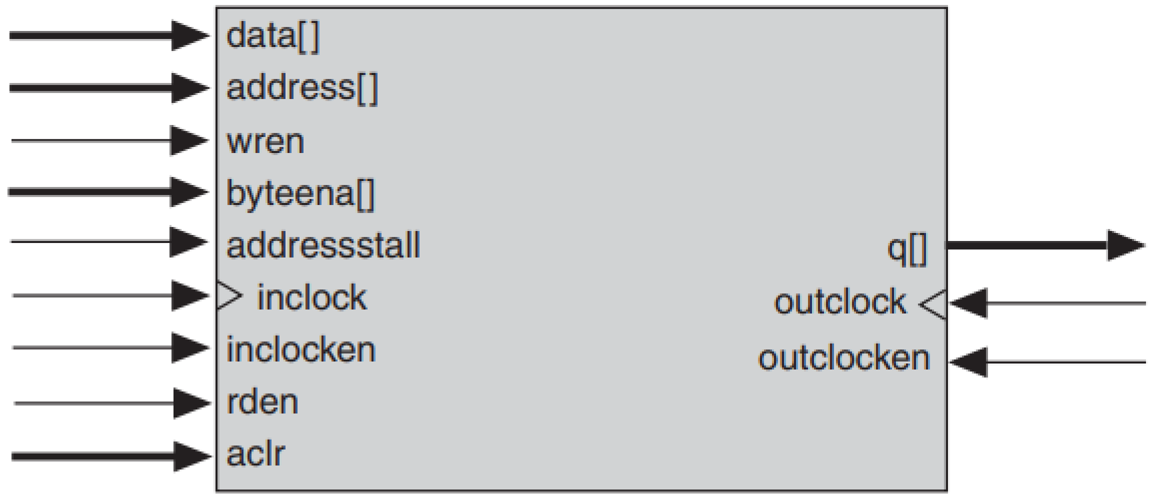
M9K memory blocks support the following modes:

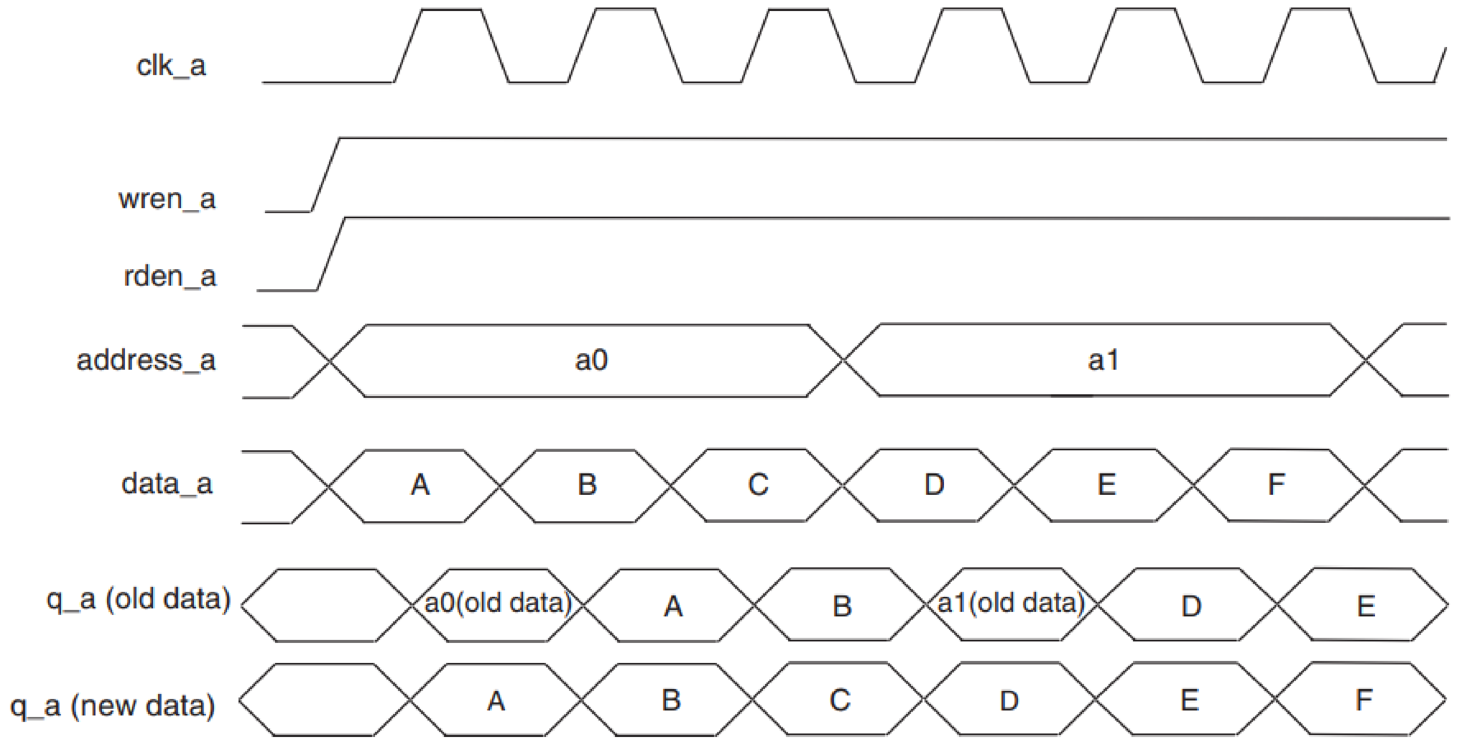
- Single-port
- Simple dual-port
- True dual-port
- Shift-register
- ROM
- FIFO

M9K implementatin

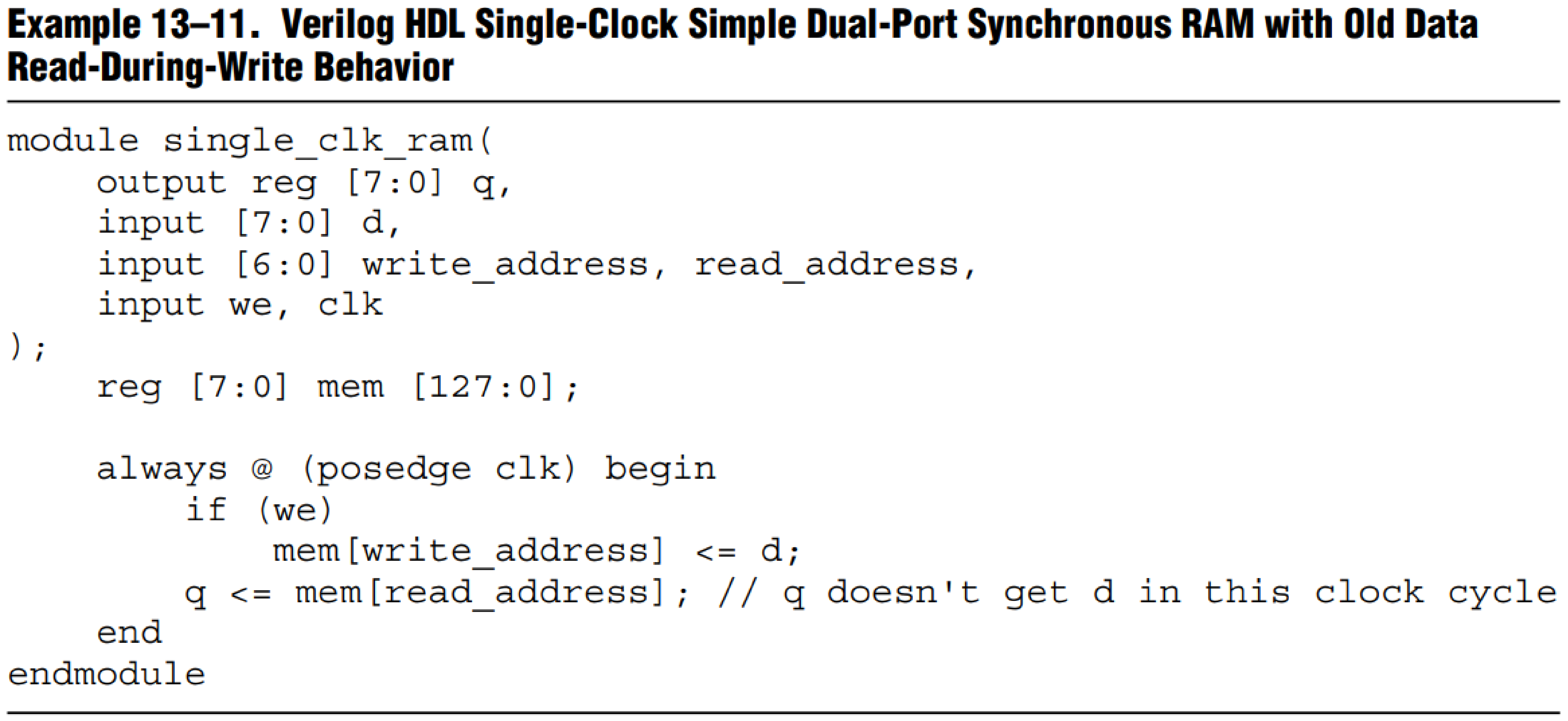
- При помощи IP каталога
- Через HDL код

M9K Single-Port





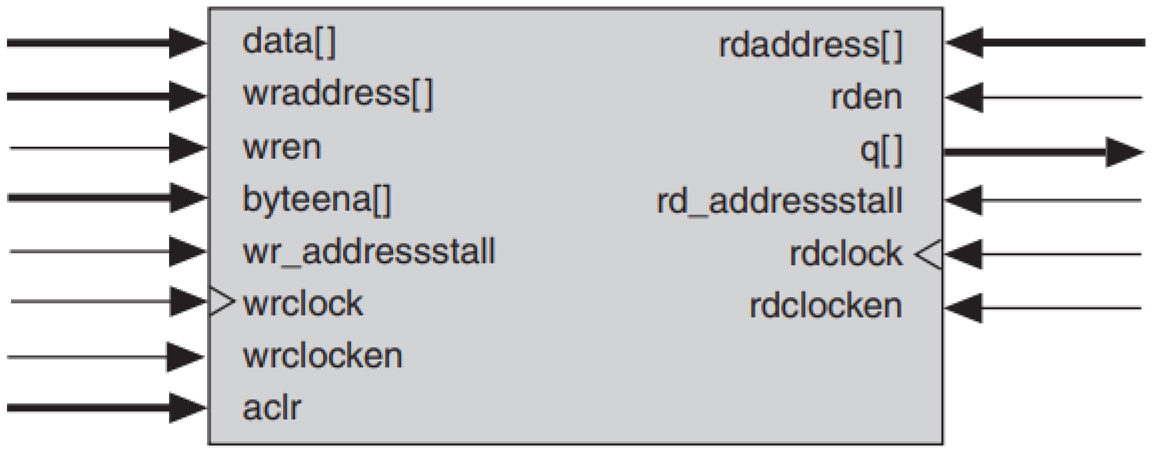
M9K Single-Port

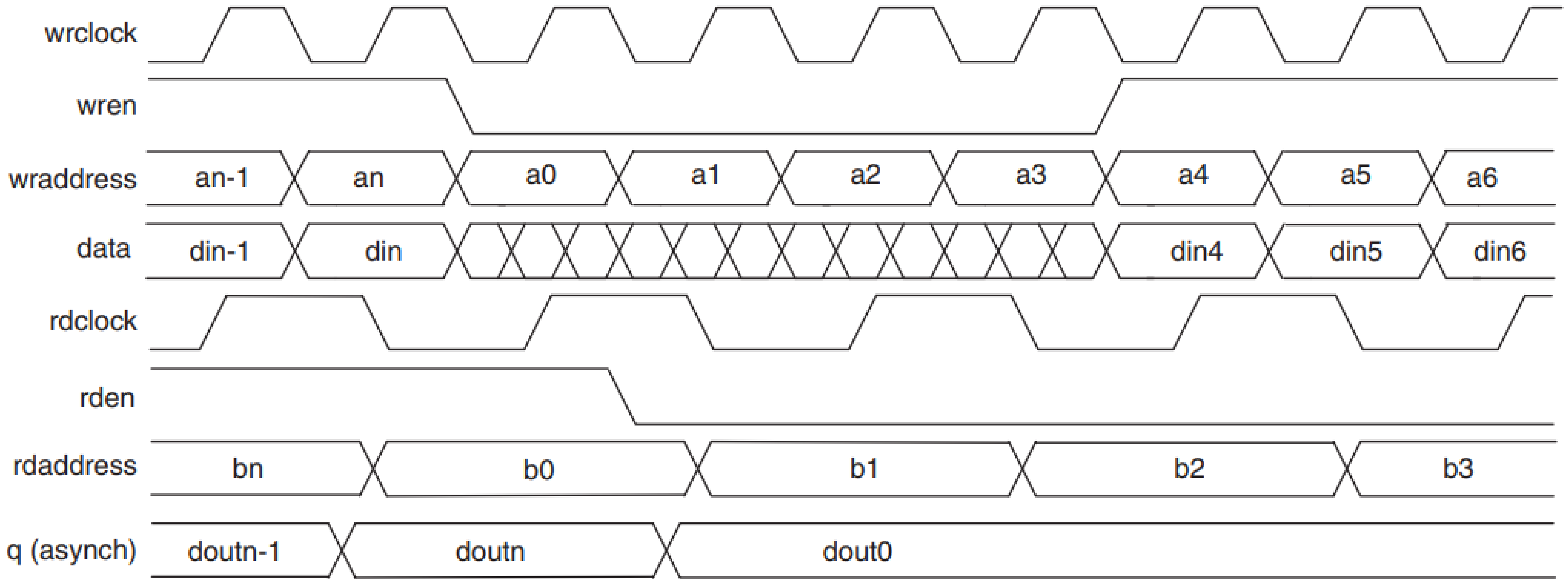


Example 13–11. Verilog HDL Single-Clock Simple Dual-Port Synchronous RAM with Old Data Read-During-Write Behavior

```
module single_clk_ram(
    output reg [7:0] q,
    input [7:0] d,
    input [6:0] write_address, read_address,
    input we, clk
);
    reg [7:0] mem [127:0];

    always @ (posedge clk) begin
        if (we)
            mem[write_address] <= d;
        q <= mem[read_address]; // q doesn't get d in this clock cycle
    end
endmodule
```

M9K Simple Dual-Port



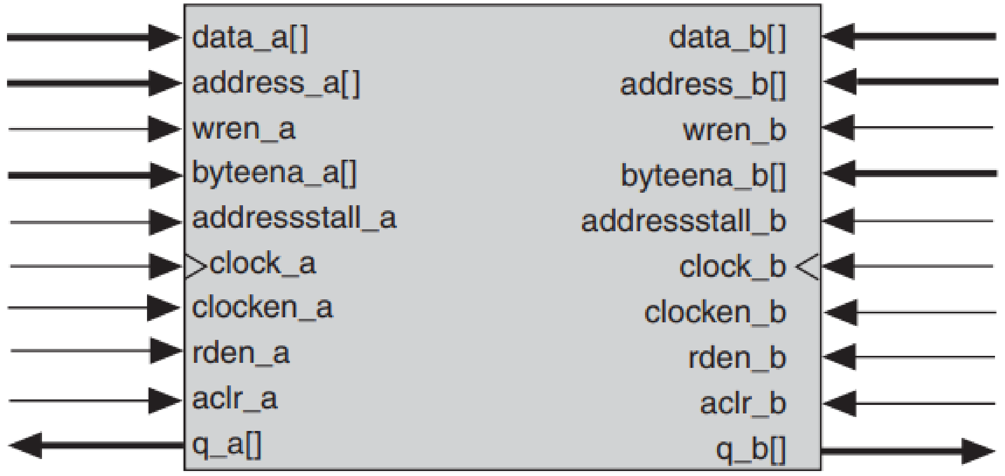


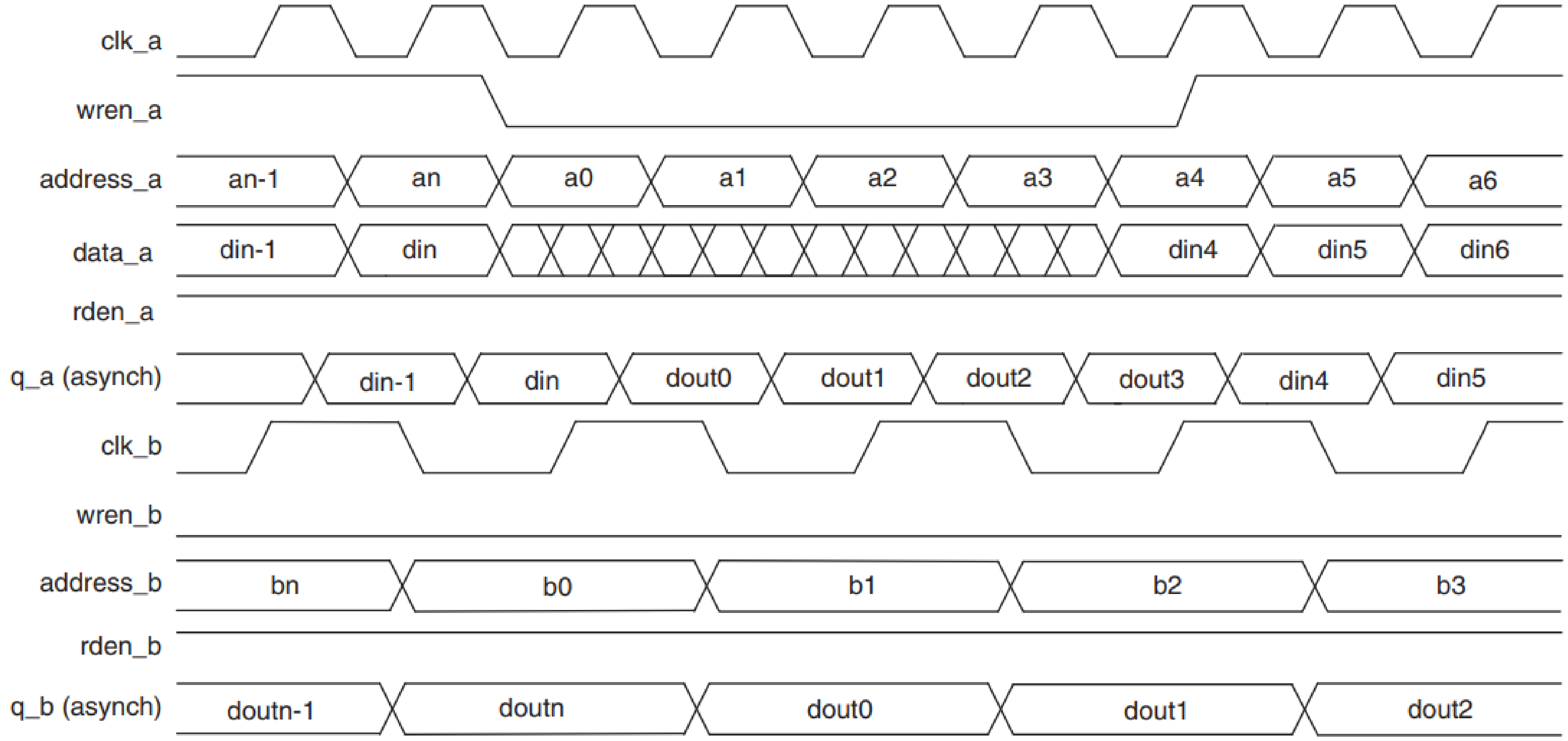
M9K Simple Dual-Port

Example 13–15. Verilog HDL Simple Dual-Port, Dual-Clock Synchronous RAM

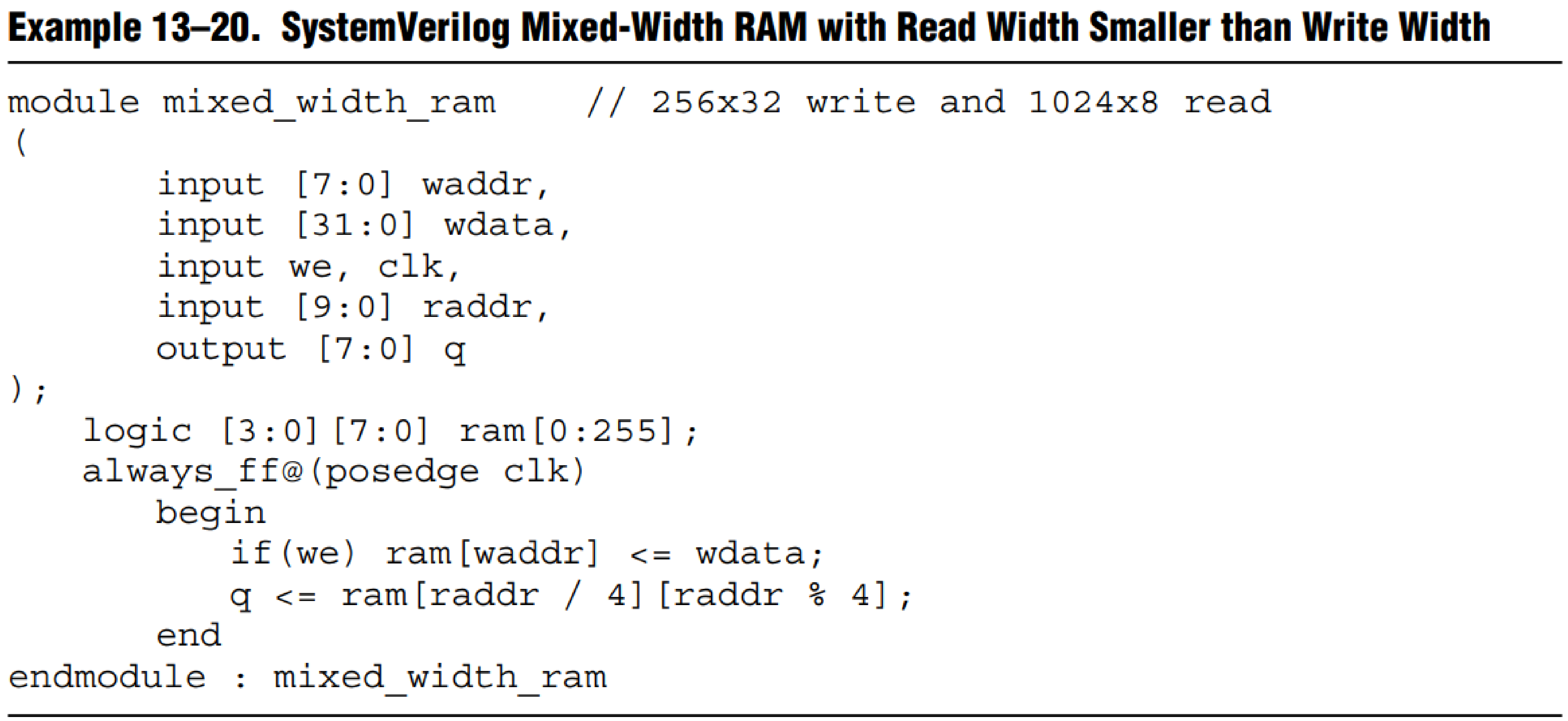
```
module dual_clock_ram(
   output reg [7:0] q,
   input [7:0] d,
   input [6:0] write_address, read_address,
   input we, clk1, clk2
);
   reg [6:0] read_address_reg;
   reg [7:0] mem [127:0];

   always @ (posedge clk1)
   begin
      if (we)
         mem[write_address] <= d;
   end

   always @ (posedge clk2) begin
      q <= mem[read_address_reg];
      read_address_reg <= read_address;
   end
endmodule
```

M9K True Dual-Port





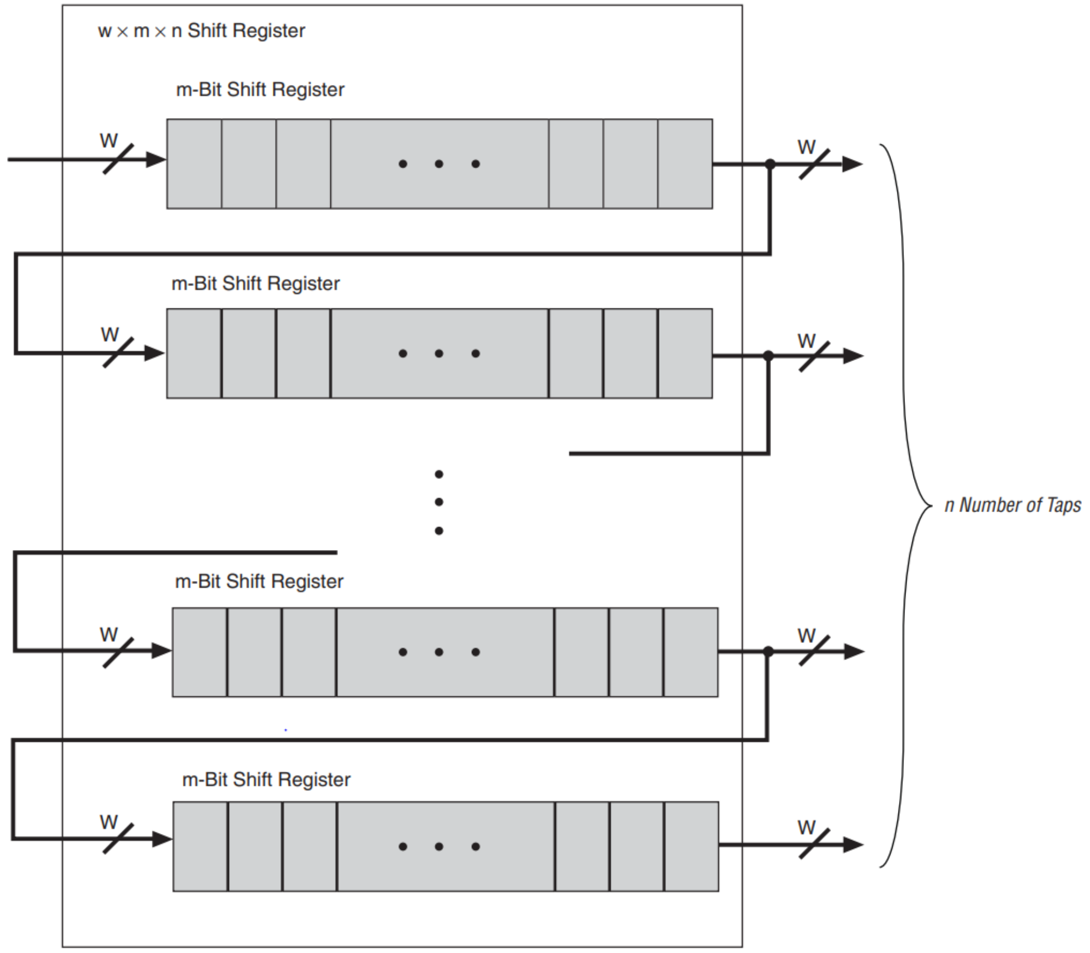
M9K True Dual-Port



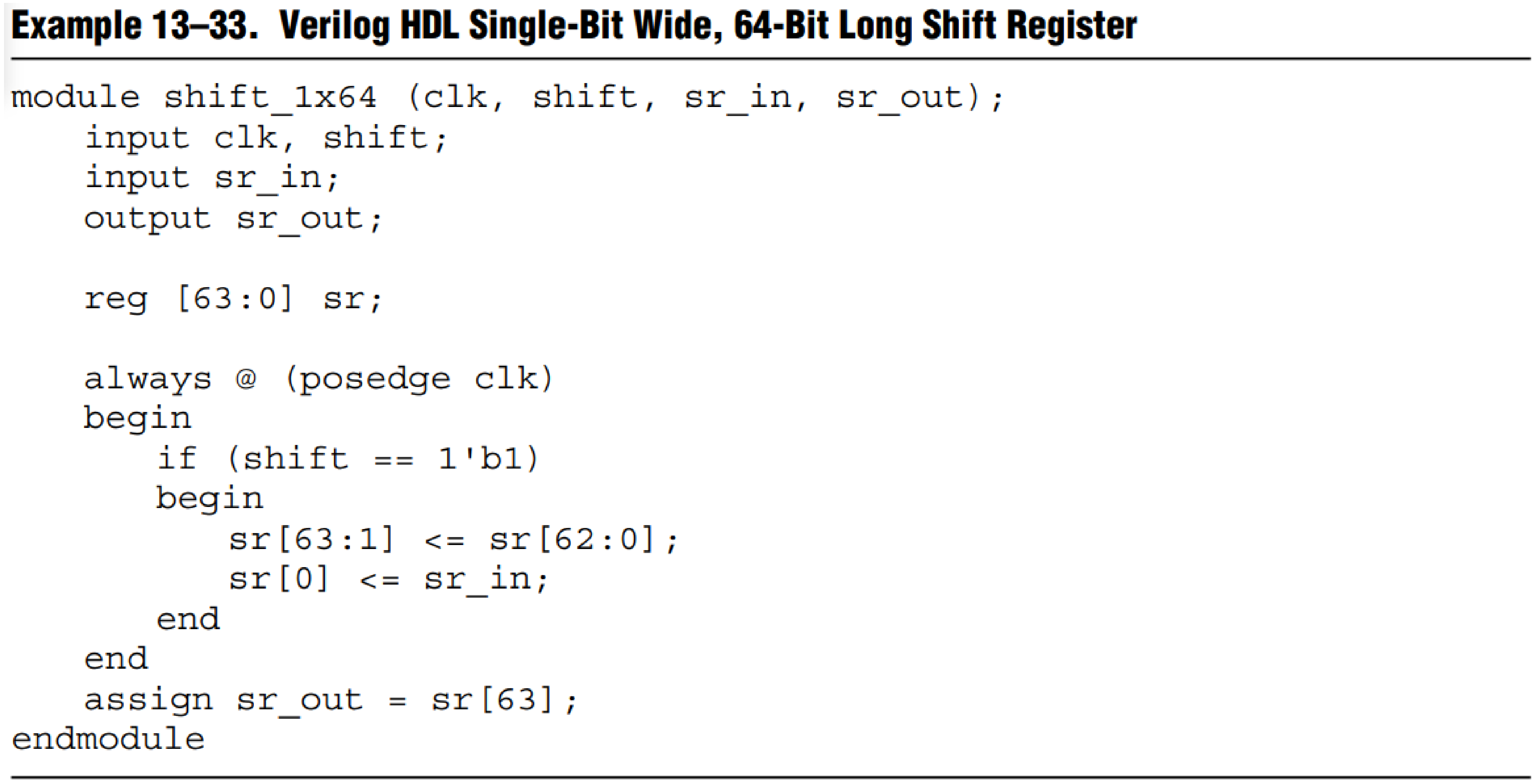
Example 13–20. SystemVerilog Mixed-Width RAM with Read Width Smaller than Write Width

```
module mixed_width_ram    // 256x32 write and 1024x8 read
(
       input [7:0] waddr,
       input [31:0] wdata,
       input we, clk,
       input [9:0] raddr,
       output [7:0] q
);
   logic [3:0][7:0] ram[0:255];
   always_ff@(posedge clk)
       begin
          if(we) ram[waddr] <= wdata;
          q <= ram[raddr / 4][raddr % 4];
       end
endmodule : mixed_width_ram
```

M9K Shift Register



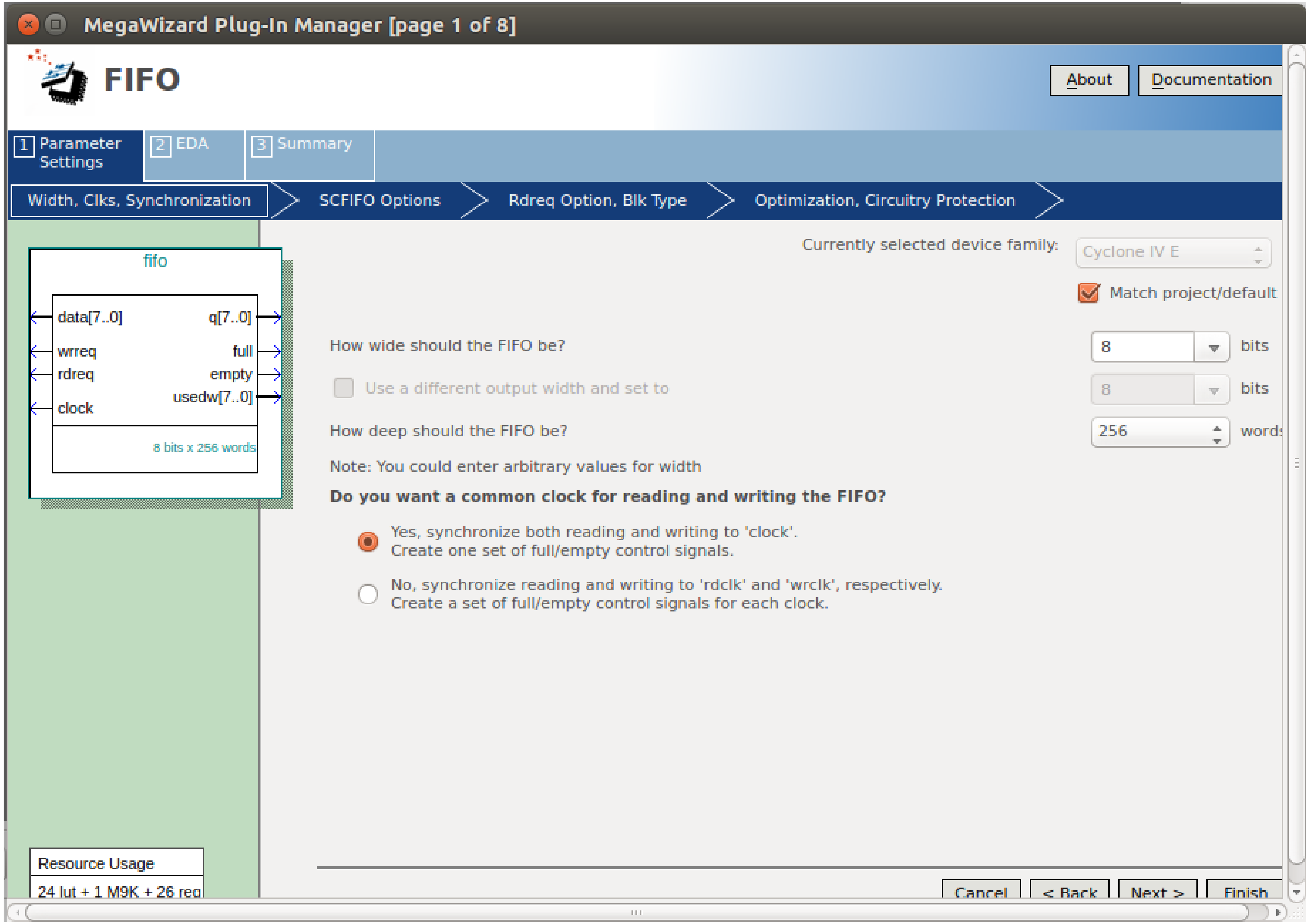
M9K Shift Register



Example 13–33. Verilog HDL Single-Bit Wide, 64-Bit Long Shift Register

```
module shift_1x64 (clk, shift, sr_in, sr_out);
    input clk, shift;
    input sr_in;
    output sr_out;

    reg [63:0] sr;

    always @ (posedge clk)
    begin
        if (shift == 1'b1)
        begin
            sr[63:1] <= sr[62:0];
            sr[0] <= sr_in;
        end
    end
    assign sr_out = sr[63];
endmodule
```

M9K FIFO



MegaWizard Plug-In Manager [page 1 of 8]

FIFO

Currently selected device family: Cyclone IV E

[✓] Match project/default

How wide should the FIFO be? 8 bits

[ ] Use a different output width and set to 8 bits

How deep should the FIFO be? 256 words

Note: You could enter arbitrary values for width

Do you want a common clock for reading and writing the FIFO?

[✓ Yes, synchronize both reading and writing to 'clock'. Create one set of full/empty control signals.]

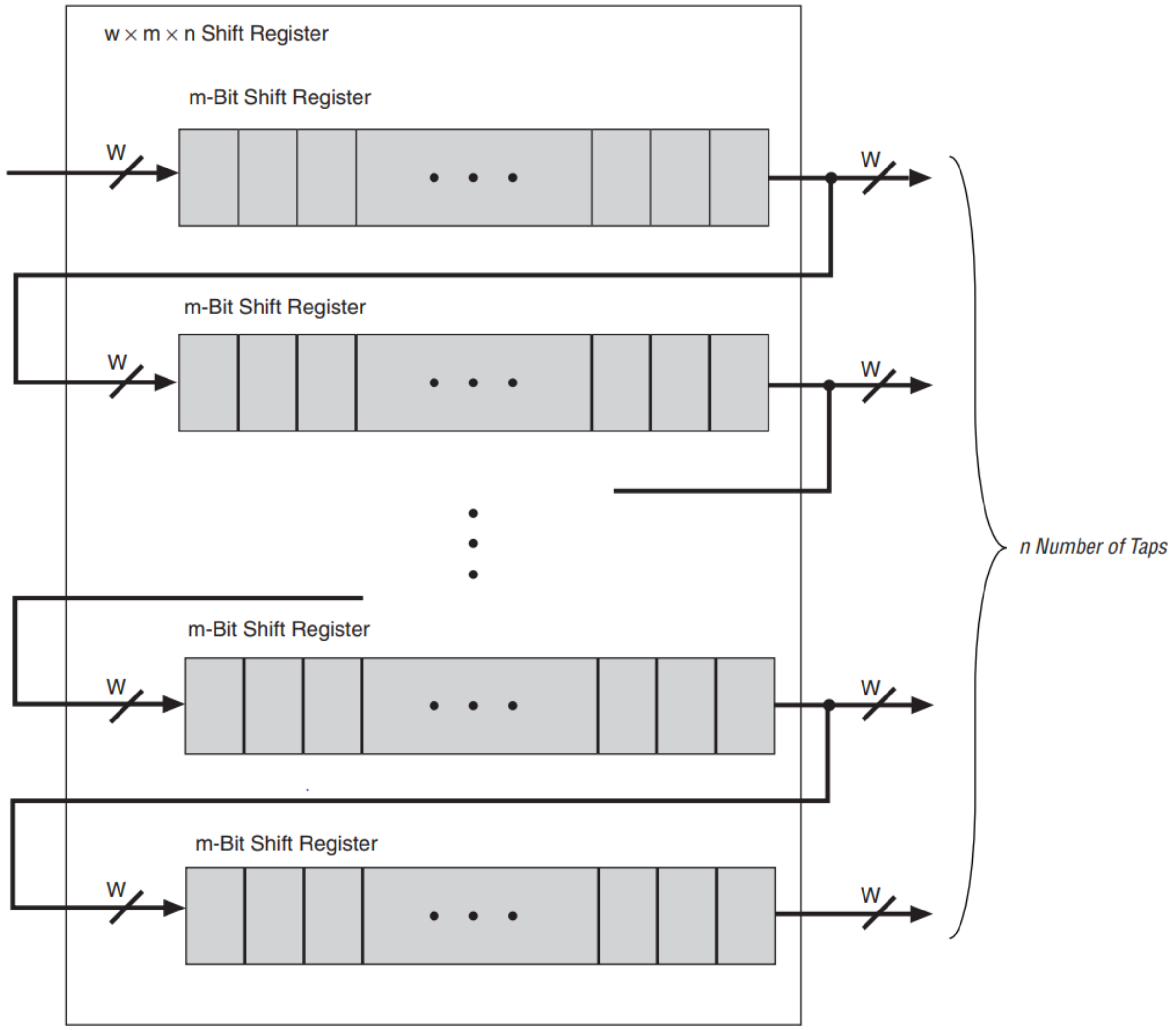
No, synchronize reading and writing to 'rdclk' and 'wrclk', respectively. Create a set of full/empty control signals for each clock.

Resource Usage

24 lut + 1 M9K + 26 reg

M9K ROM

- ALTSHIFT_TAPS MegaFunction

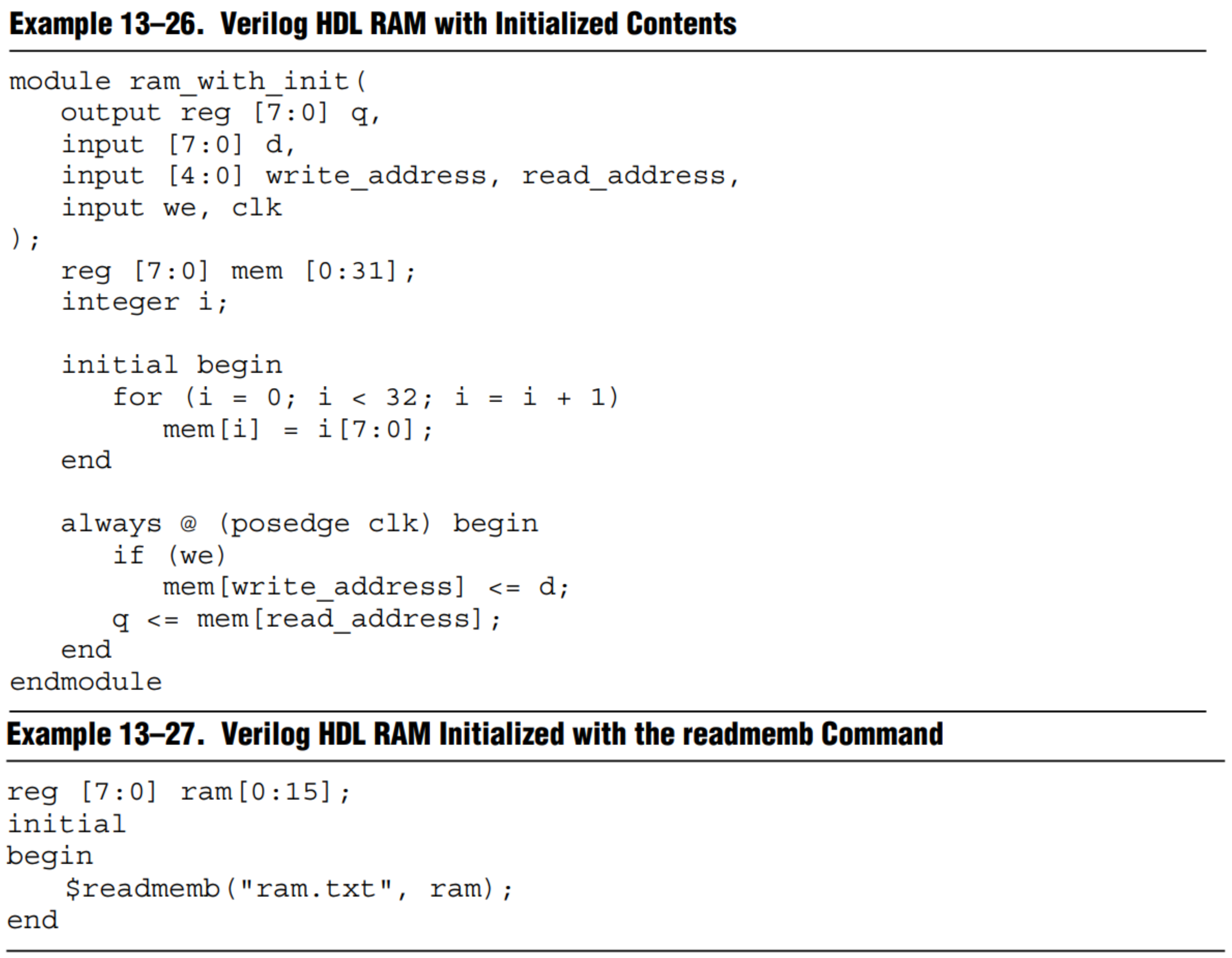


M9K ROM

Example 13–31. Verilog HDL Dual-Port Synchronous ROM Using readmemb

```
module dual_port_rom (
   input [(addr_width-1):0] addr_a, addr_b,
   input clk,
   output reg [(data_width-1):0] q_a, q_b
);
   parameter data_width = 8;
   parameter addr_width = 8;

   reg [data_width-1:0] rom[2**addr_width-1:0];

   initial // Read the memory contents in the file
          //dual_port_rom_init.txt.
   begin
      $readmemb("dual_port_rom_init.txt", rom);
   end

   always @ (posedge clk)
   begin
      q_a <= rom[addr_a];
      q_b <= rom[addr_b];
   end
endmodule
```

Memory Initialization



Example 13–26. Verilog HDL RAM with Initialized Contents

```
module ram_with_init(
   output reg [7:0] q,
   input [7:0] d,
   input [4:0] write_address, read_address,
   input we, clk
);
   reg [7:0] mem [0:31];
   integer i;

   initial begin
      for (i = 0; i < 32; i = i + 1)
         mem[i] = i[7:0];
   end

   always @ (posedge clk) begin
      if (we)
         mem[write_address] <= d;
      q <= mem[read_address];
   end
endmodule
```

Example 13–27. Verilog HDL RAM Initialized with the readmemb Command

```
reg [7:0] ram[0:15];
initial
begin
   $readmemb("ram.txt", ram);
end
```

Атрибуты и параметры

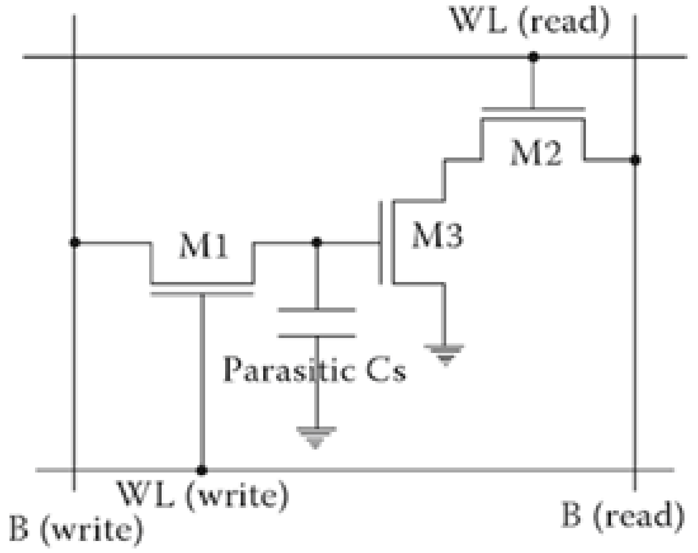
```
(* ramstyle = "M144K" *) reg [0:7] my_ram[0:63];
```

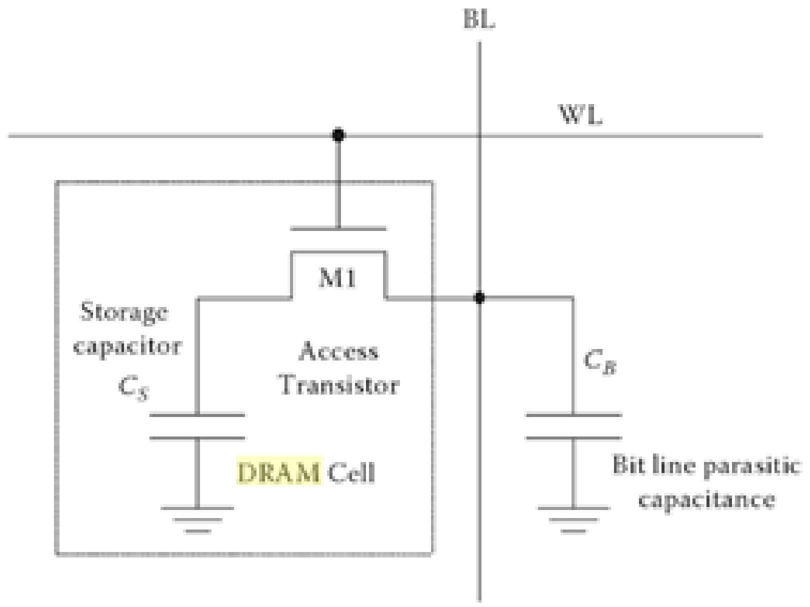
```
reg [0:7] my_ram[0:63] /* synthesis ramstyle = "M144K" */;
```

- no_rw_check
- M9K
- logic

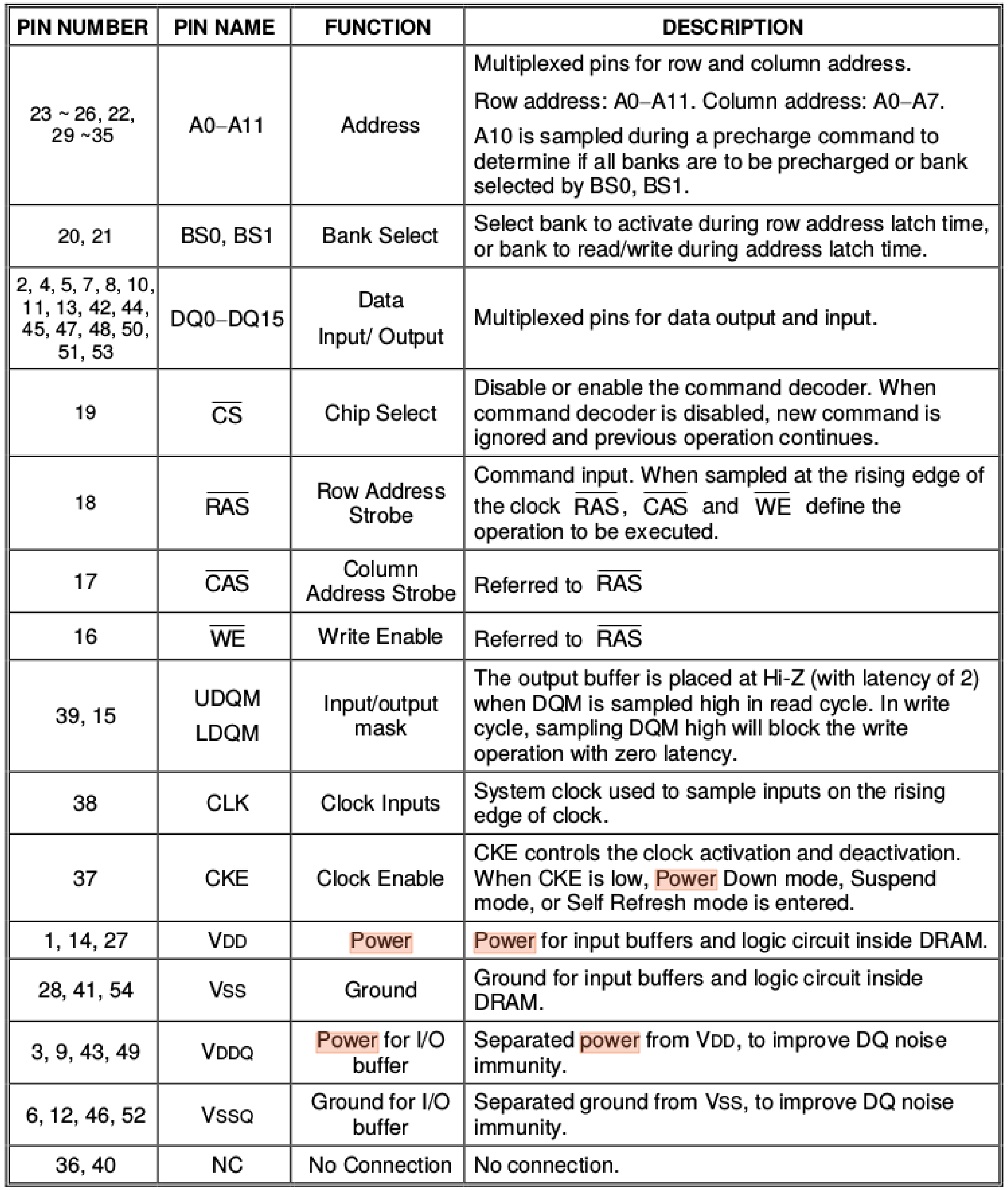
Внешняя SDRAM

- 8Mb
- Энергозависимая
- Интерфейс доступа специфицирован
-



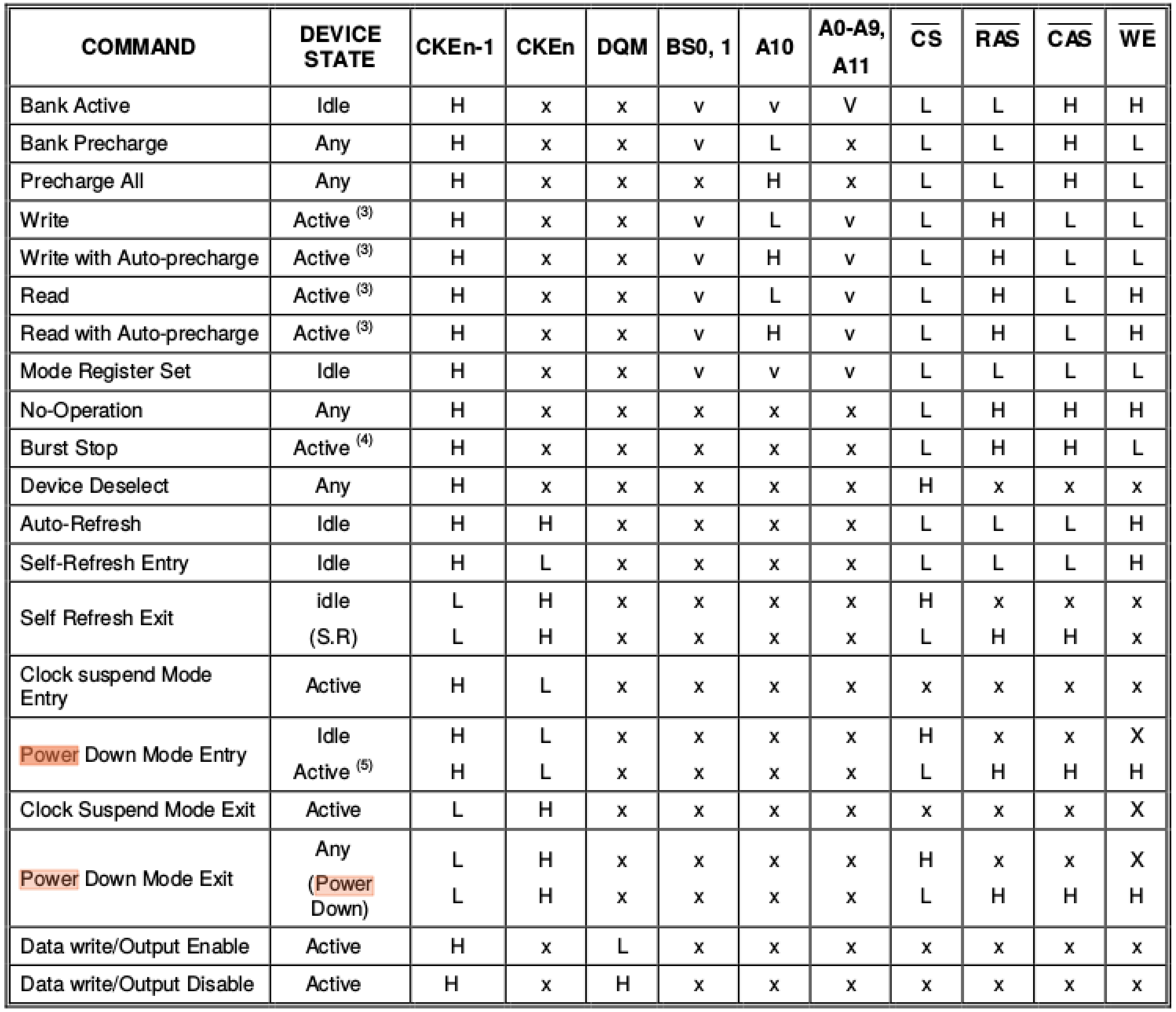


Внешняя SDRAM



PIN NUMBER	PIN NAME	FUNCTION	DESCRIPTION
23 ~ 26, 22, 29 ~35	A0–A11	Address	Multiplexed pins for row and column address. Row address: A0–A11. Column address: A0–A7. A10 is sampled during a precharge command to determine if all banks are to be precharged or bank selected by BS0, BS1.
20, 21	BS0, BS1	Bank Select	Select bank to activate during row address latch time, or bank to read/write during address latch time.
2, 4, 5, 7, 8, 10, 11, 13, 42, 44, 45, 47, 48, 50, 51, 53	DQ0–DQ15	Data Input/ Output	Multiplexed pins for data output and input.
19	$\overline{CS}$	Chip Select	Disable or enable the command decoder. When command decoder is disabled, new command is ignored and previous operation continues.
18	$\overline{RAS}$	Row Address Strobe	Command input. When sampled at the rising edge of the clock $\overline{RAS}$, $\overline{CAS}$ and $\overline{WE}$ define the operation to be executed.
17	$\overline{CAS}$	Column Address Strobe	Referred to $\overline{RAS}$
16	$\overline{WE}$	Write Enable	Referred to $\overline{RAS}$
39, 15	UDQM LDQM	Input/output mask	The output buffer is placed at Hi-Z (with latency of 2) when DQM is sampled high in read cycle. In write cycle, sampling DQM high will block the write operation with zero latency.
38	CLK	Clock Inputs	System clock used to sample inputs on the rising edge of clock.
37	CKE	Clock Enable	CKE controls the clock activation and deactivation. When CKE is low, Power Down mode, Suspend mode, or Self Refresh mode is entered.
1, 14, 27	VDD	Power	Power for input buffers and logic circuit inside DRAM.
28, 41, 54	VSS	Ground	Ground for input buffers and logic circuit inside DRAM.
3, 9, 43, 49	VDDQ	Power for I/O buffer	Separated power from VDD, to improve DQ noise immunity.
6, 12, 46, 52	VSSQ	Ground for I/O buffer	Separated ground from VSS, to improve DQ noise immunity.
36, 40	NC	No Connection	No connection.

Внешняя SDRAM



COMMAND	DEVICE STATE	CKEn-1	CKEn	DQM	BS0, 1	A10	A0-A9, A11	$\overline{CS}$	$\overline{RAS}$	$\overline{CAS}$	$\overline{WE}$
Bank Active	Idle	H	x	x	v	v	V	L	L	H	H
Bank Precharge	Any	H	x	x	v	L	x	L	L	H	L
Precharge All	Any	H	x	x	x	H	x	L	L	H	L
Write	Active [3]	H	x	x	v	L	v	L	H	L	L
Write with Auto-precharge	Active [3]	H	x	x	v	H	v	L	H	L	L
Read	Active [3]	H	x	x	v	L	v	L	H	L	H
Read with Auto-precharge	Active [3]	H	x	x	v	H	v	L	H	L	H
Mode Register Set	Idle	H	x	x	v	v	v	L	L	L	L
No-Operation	Any	H	x	x	x	x	x	L	H	H	H
Burst Stop	Active [4]	H	x	x	x	x	x	L	H	H	L
Device Deselect	Any	H	x	x	x	x	x	H	x	x	x
Auto-Refresh	Idle	H	H	x	x	x	x	L	L	L	H
Self-Refresh Entry	Idle	H	L	x	x	x	x	L	L	L	H
Self Refresh Exit	idle	L	H	x	x	x	x	H	x	x	x
	(S.R)	L	H	x	x	x	x	L	H	H	x
Clock suspend Mode Entry	Active	H	L	x	x	x	x	x	x	x	x
Power Down Mode Entry	Idle	H	L	x	x	x	x	H	x	x	X
	Active [5]	H	L	x	x	x	x	L	H	H	H
Clock Suspend Mode Exit	Active	L	H	x	x	x	x	x	x	x	X
Power Down Mode Exit	Any	L	H	x	x	x	x	H	x	x	X
	(Power Down)	L	H	x	x	x	x	L	H	H	H
Data write/Output Enable	Active	H	x	L	x	x	x	x	x	x	x
Data write/Output Disable	Active	H	x	H	x	x	x	x	x	x	x