

Белорусский государственный университет
информатики и радиоэлектроники

4-ый курс специальности ПОИТ

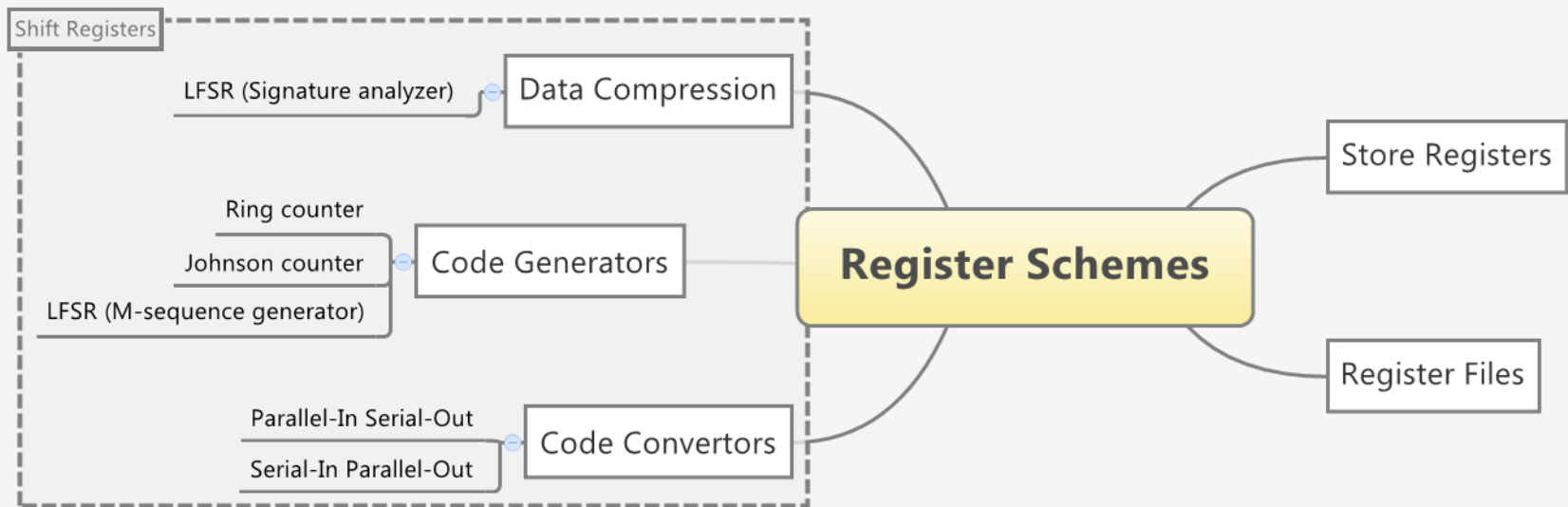
ПРОГРАММНОЕ ОБЕСПЕЧЕНИЕ ЦИФРОВОГО ПРОЕКТИРОВАНИЯ



2020

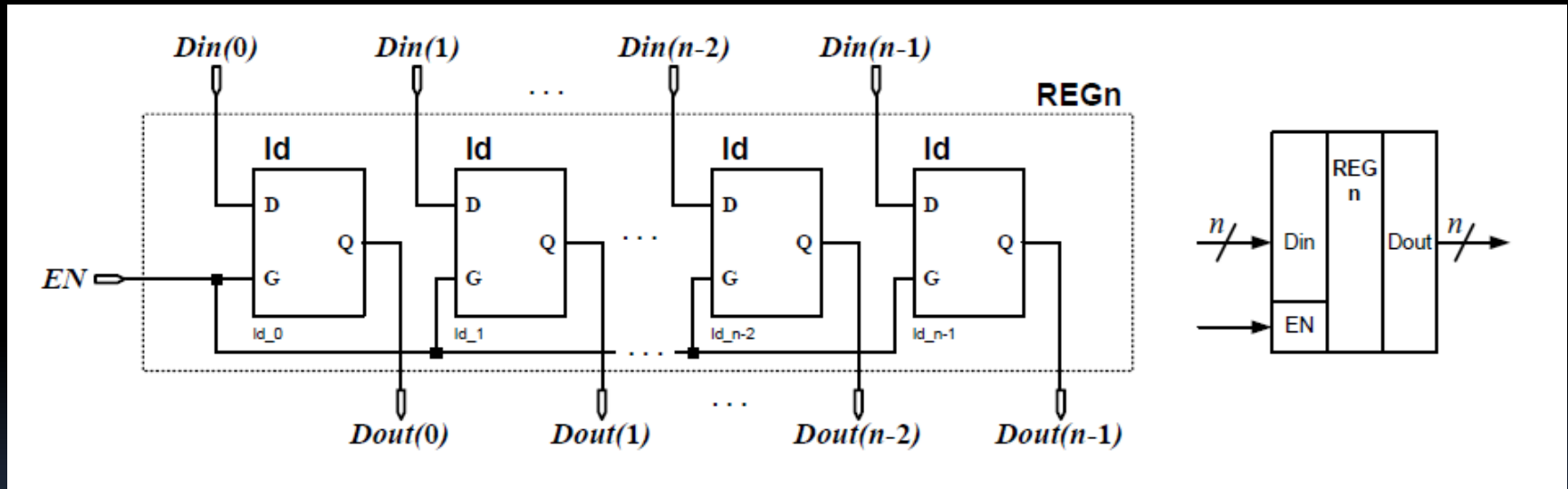
Introducing to VHDL

Sequential Logic: registers



Introducing to VHDL

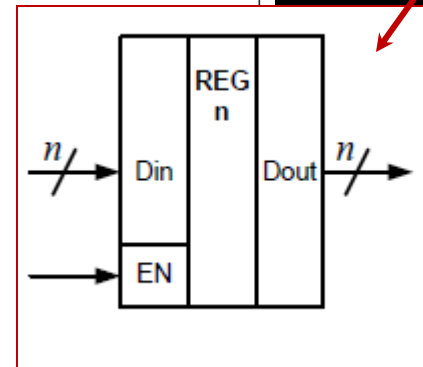
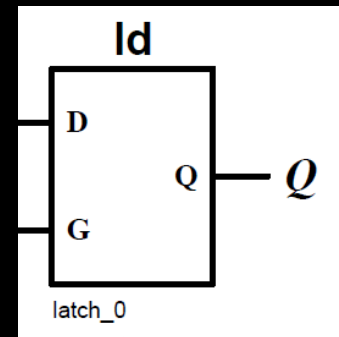
Sequential Logic: store registers



Introducing to VHDL

Sequential Logic: store registers

```
1 Entity REGn is
2   Generic
3     — разрядность хранимых слов
4     ( n      : integer := 4 );
5   Port ( — входная шина данных
6         Din   : in  std_logic_vector( n-1 downto 0 );
7         — входной порт сигнала загрузки данных
8         EN    : in  std_logic;
9         — выходная шина данных
10        Dout  : out std_logic_vector( n-1 downto 0 );
11 End REGn;
12
13 Architecture Beh of REGn is
14   — множество элементов памяти мощностью n
15   signal reg : std_logic_vector( n-1 downto 0 );
16
17 Begin
18
19   Main: process( Din, EN )
20   begin
21     — условие захвата данных с входной шины Din
22     if EN = '1' then
23       reg <= Din;
24     — при всех прочих условиях осуществляется
25     — хранение n-разрядного слова
26     end if;
27   end process;
28
29   — передача значения хранящегося слова
30   — на выходную шину Dout
31   Dout <= reg;
32
33 End Beh;
```



Introducing to VHDL

Sequential Logic: synchronous registers

```
1  Entity REGn is
2      Generic
3          ( — константное слово для инициализации регистра,
4            — также определяет размерность регистра,
5            — по умолчанию константное слово равно "1001",
6            — размерность n=4
7            INITREG : std_logic_vector := "1001" );
8
9      Port ( — входная шина данных
10          Din      : in  std_logic_vector( INITREG'range );
11          — входной порт сигнала загрузки данных
12          EN       : in  std_logic;
13          — входной порт сигнала инициализации
14          INIT     : in  std_logic;
15          — входной порт сигнала синхронизации,
16          — управление передним фронтом
17          CLK      : in  std_logic;
18          — входной порт сигнала управления выходными
19          — трехстабильными буферами
20          OE       : in  std_logic;
21          — выходная шина данных
22          Dout     : out std_logic_vector( INITREG'range );
23  End REGn;
24
```

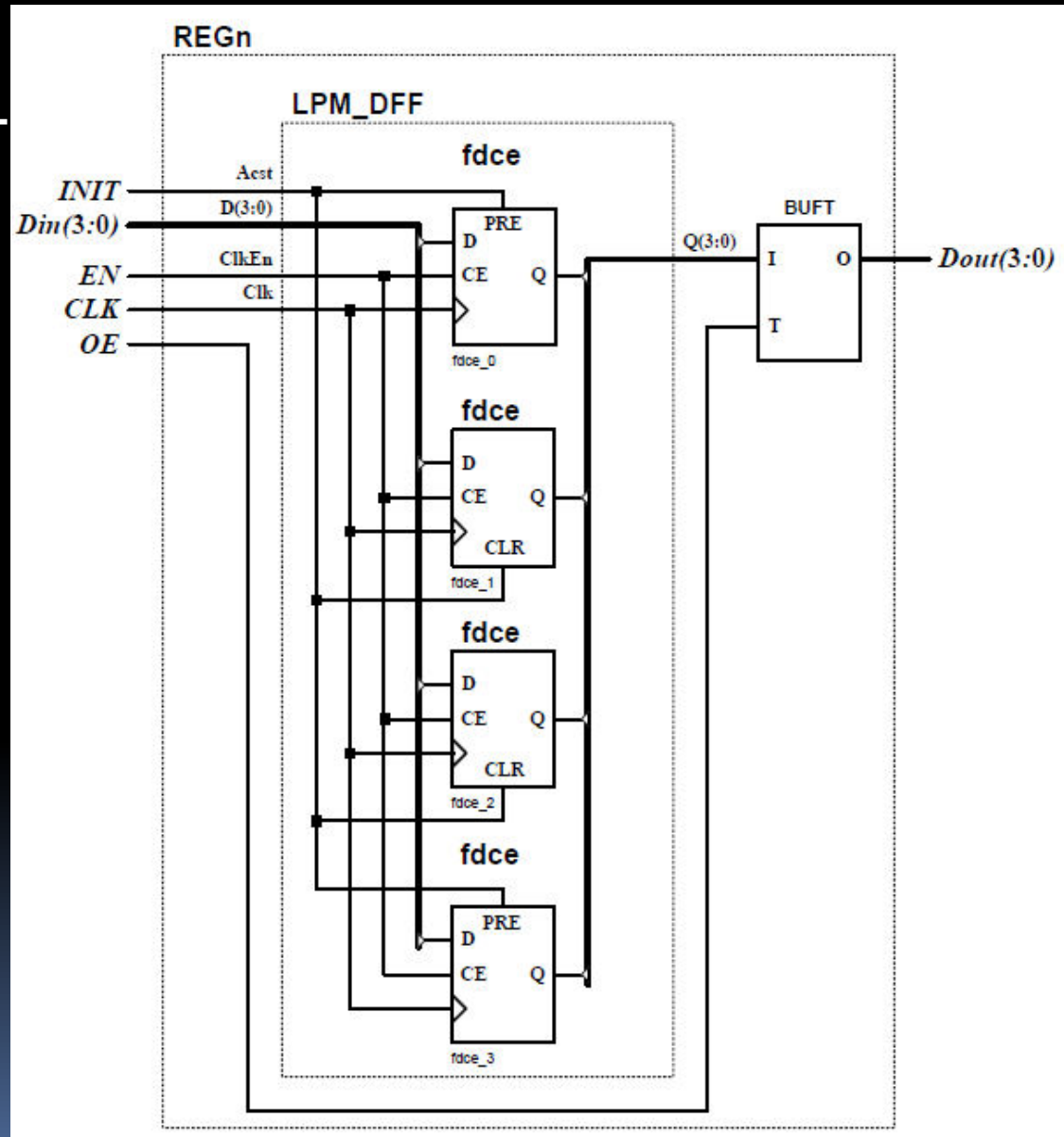
Introducing to VHDL

Sequential Logic: synchronous registers

```
25 Architecture Beh of REGn is
26 — непосредственно сам регистр (синхронные D-триггеры)
27 signal reg      : std_logic_vector( INITREG'range );
28 — константный вектор "ZZZ...Z" размерностью INITREG'range
29 constant ALLZ   : std_logic_vector( INITREG'range ) := ( others => 'Z' );
30
31 Begin
32
33   Main: process( Din , EN, INIT , CLK )
34   begin
35     — условие инициализации
36     if INIT = '1' then
37       reg <= INITREG;
38     — условие разрешения загрузки данных
39     elsif EN = '1' then
40       — условие наступления переднего фронта
41       — сигнала синхронизации
42       if rising_edge( CLK ) then
43         — захват и загрузка данных
44         reg <= Din;
45       end if;
46     end if;
47   end process;
48
49   — выходные трехстабильные буферы-усилители
50   Dout <= reg when OE='0' else ALLZ;
51
52 End Beh;
```

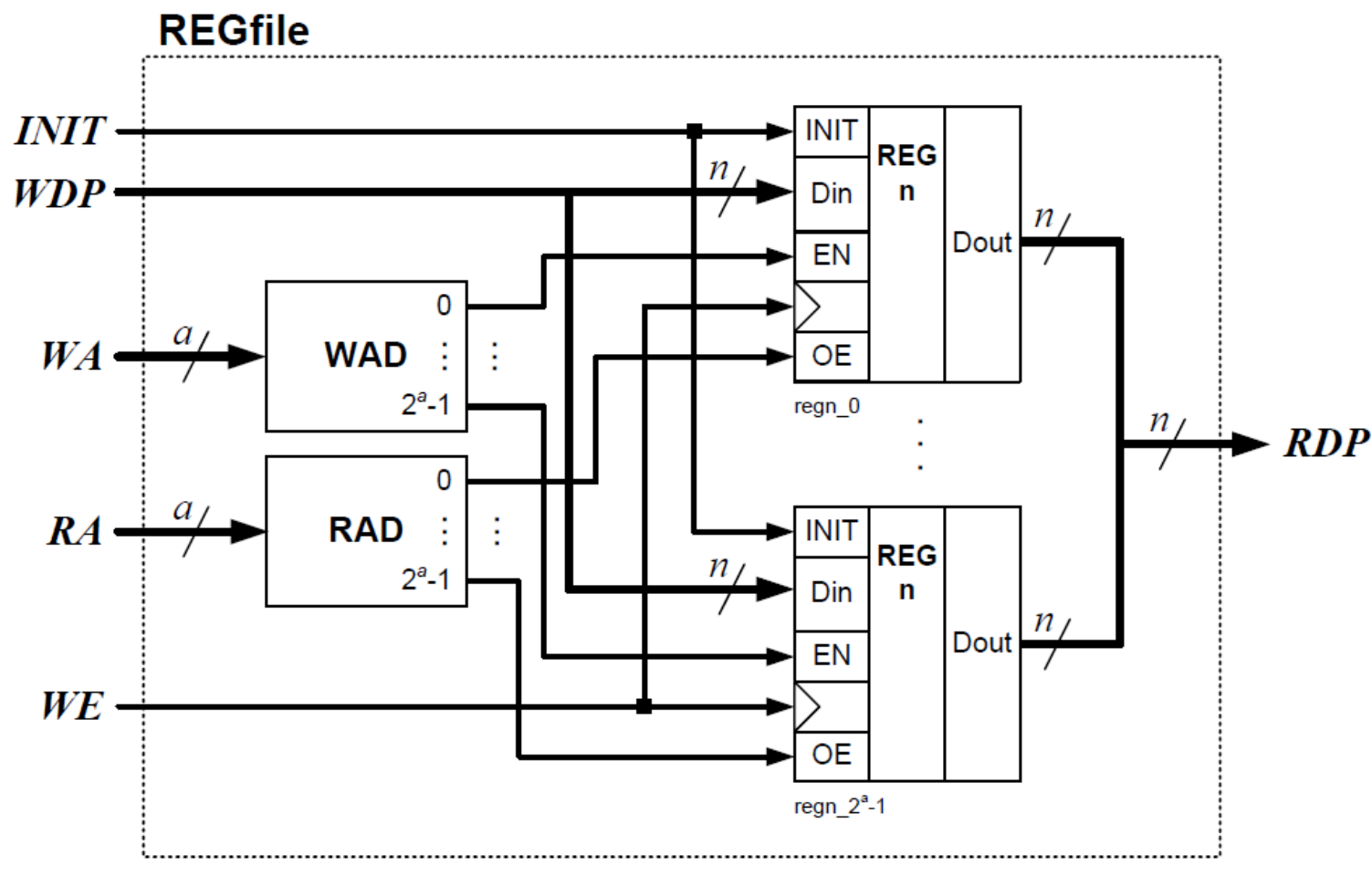
Introducing to VHDL

Sequential Logic: synchronous registers



Introducing to VHDL

Sequential Logic: register files



Introducing to VHDL

Sequential Logic: register files

```
1  Entity REGFile is
2    Generic
3      (  — константное слово для инициализации регистров,
4        — также определяет размерность регистров,
5        — по умолчанию константное слово равно "0000",
6        — размерность n=4
7        INITREG : std_logic_vector := "0000";
8        — размерности адресных шин записи и чтения
9        — для адресации 2**a регистров
10       a       : integer := 2 );
11    Port ( — входной порт сигнала инициализации регистров
12          INIT   : in std_logic;
13          — входной порт шины данных для записи
14          WDP     : in std_logic_vector( INITREG'range );
15          — входной порт шины адреса для записи
16          WA      : in std_logic_vector( a-1 downto 0 );
17          — входной порт шины адреса для чтения
18          RA      : in std_logic_vector( a-1 downto 0 );
19          — входной порт сигнала разрешения записи
20          WE      : in std_logic;
21          — выходной порт шины прочитанных данных
22          RDP     : out std_logic_vector( INITREG'range );
23  End REGFile;
24
```

Introducing to VHDL

Sequential Logic: register files

```
25 Architecture Beh of REGFile is
26 — декларация компоненты синхронного регистра хранения
27 component REGn is
28   Generic
29     ( INITREG : std_logic_vector := "1001" );
30
31   Port ( Din      : in   std_logic_vector( INITREG'range );
32         EN        : in   std_logic;
33         INIT      : in   std_logic;
34         CLK       : in   std_logic;
35         OE        : in   std_logic;
36         Dout      : out  std_logic_vector( INITREG'range );
37 end component;
38 — выходная шина дешифратора адреса записи
39 signal wen  : std_logic_vector( 2**a-1 downto 0 );
40 — выходная шина дешифратора адреса чтения
41 signal ren  : std_logic_vector( 2**a-1 downto 0 );
42 — общая шина прочитанных данных
43 signal readd: std_logic_vector( INITREG'range );
44
```

Introducing to VHDL

Sequential Logic: register files (mixed style)

```
45 Begin
46
47 — дешифратор адреса записи
48 WAD: process( WA )
49 begin
50   for i in 0 to 2**a-1 loop
51     if i = CONV_INTEGER( WA ) then
52       wen( i ) <= '1';
53     else
54       wen( i ) <= '0';
55     end if;
56   end loop;
57 end process;
```

Combinational Logic
Behavioral Description

```
59 — дешифратор адреса чтения
60 RAD: process( RA )
61 begin
62   for i in 0 to 2**a-1 loop
63     if i = CONV_INTEGER( RA ) then
64       ren( i ) <= '1';
65     else
66       ren( i ) <= '0';
67     end if;
68   end loop;
69 end process;
```

Combinational Logic
Behavioral Description

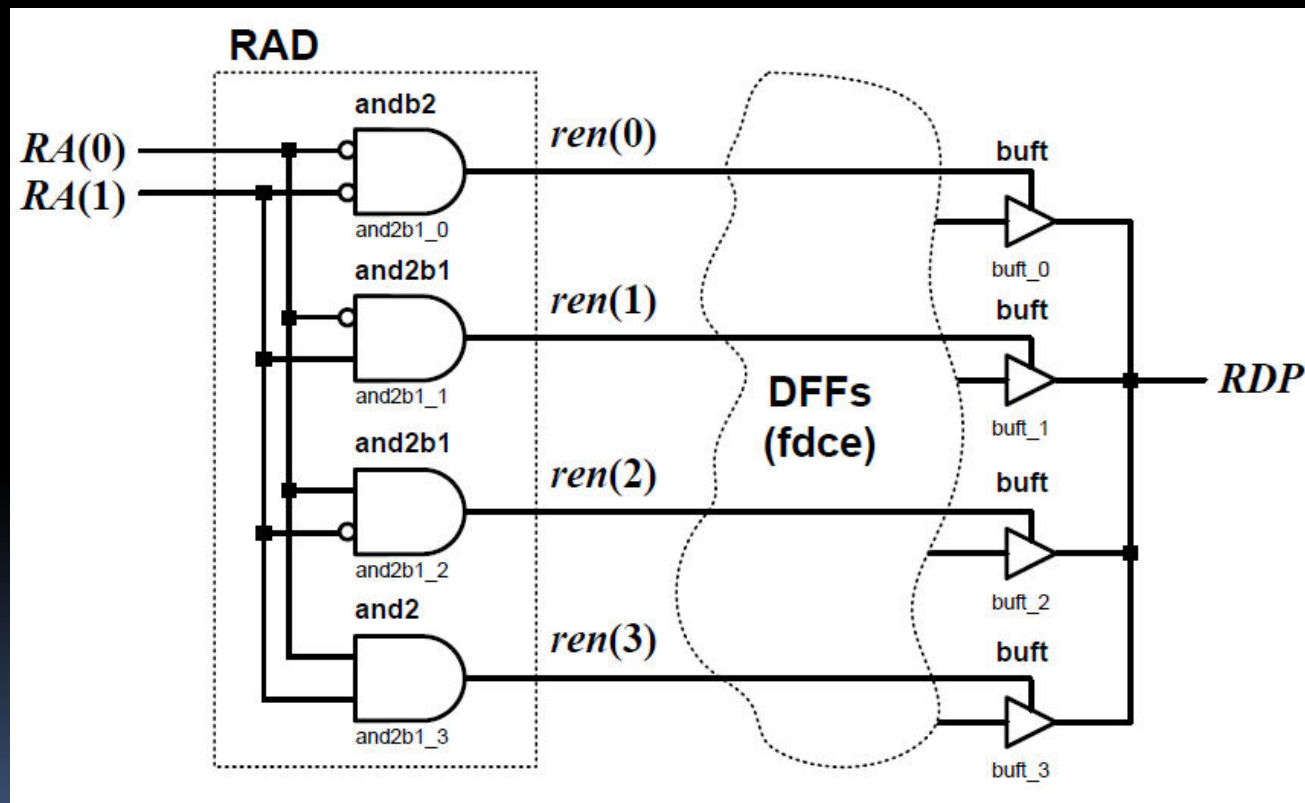
```
71 — множество из 2**a регистров REGn
72 REGi: for i in 2**a-1 downto 0 generate
73   REGi: REGn generic map ( INITREG )
74     port map ( WDP, wen(i), INIT, WE, ren(i), readd );
75 end generate;
```

Sequential Logic
Structural Description

```
76
77 — трансляция общей шины прочитанных данных на выходной порт RDP
78 RDP <= readd;
79
80 End Beh;
```

Introducing to VHDL

Sequential Logic: register files (addressing schemes)



Introducing to VHDL

Sequential Logic: register files (addressing schemes)

