

ΟΜΑΔΑ 10

ΔΑΣΟΥΛΑΣ ΙΩΑΝΝΗΣ 1053711

ΜΑΡΓΑΡΙΤΗΣ ΓΡΗΓΟΡΙΟΣ 1053651

ΑΣΚΗΣΗ 1

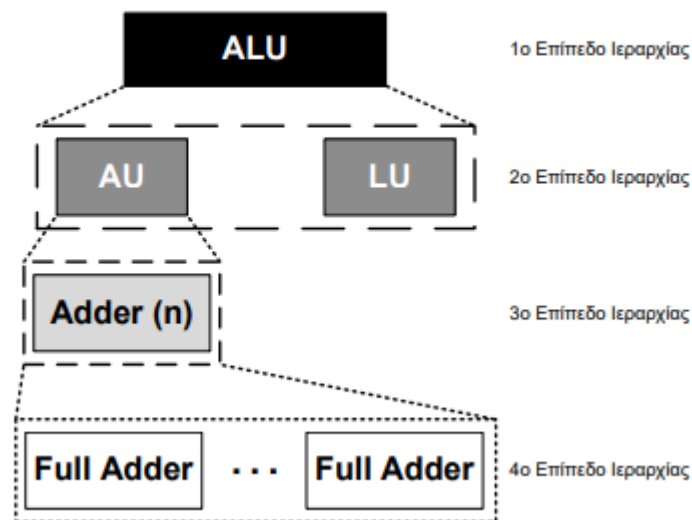
Σχεδίαση και Υλοποίηση μίας ALU

Στόχος της άσκησης είναι αρχικά να σχεδιαστεί μια πλήρως λειτουργική μονάδα αριθμητικών και λογικών πράξεων σε δομική σχεδιαστική μορφή. Η μονάδα θα πρέπει να έχει τις εξής εισόδους και εξόδους: A: Είσοδος (8-bit) – Πρώτος τελεστής σε συμπλήρωμα ως προς 2 B: Είσοδος (8-bit) – Δεύτερος τελεστής σε συμπλήρωμα ως προς 2 Op: Είσοδος (3-bit) – Κωδικός πράξης Out: Έξοδος (8-bit) – Αποτέλεσμα σε συμπλήρωμα ως προς 2. Zero: Έξοδος (1-bit) – Ενεργοποιημένη αν το αποτέλεσμα είναι μηδέν Cout: Έξοδος (1-bit) – Ενεργοποιημένη αν υπήρξε κρατούμενο (Carry)

Κωδικός	Πράξη	Αποτέλεσμα
Op = 000	Πρόσθεση	Out = A + B
Op = 001	Αφαίρεση	Out = A - B
Op = 100	Λογικό «ΚΑΙ»	Out = A & B
Op = 101	Αντιστροφή του A	Out = ! A
Op = 110	Λογικό «Η»	Out = A B
Op = 111	Λογικό «XOR»	Out = A \oplus B

Για την υλοποίηση της ALU έπρεπε να χρησιμοποιήσουμε 4 επίπεδα ιεραρχίας σύμφωνα με το παρακάτω σχήμα

Ιεραρχία της σχεδίασης:



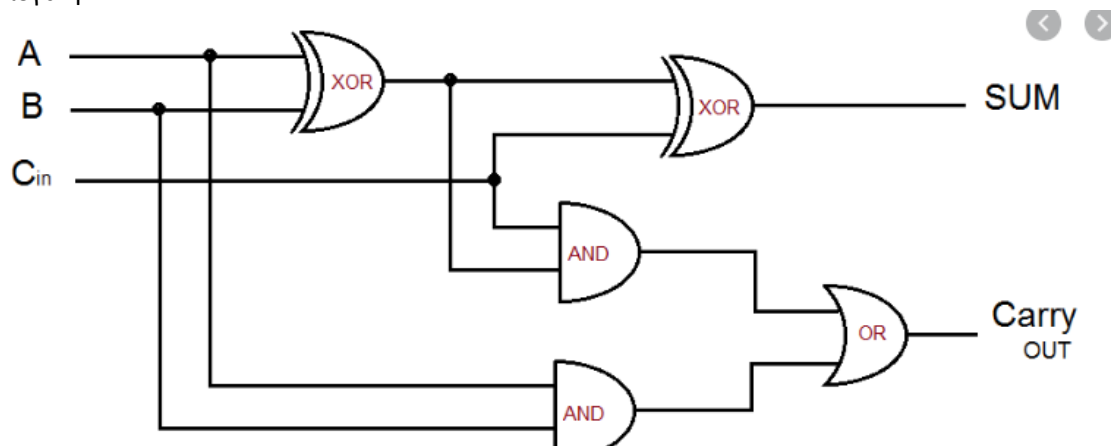
Για αυτό χρησιμοποιήθηκαν 5 αρχεία .vhd όπως φαίνεται στην παρακάτω φωτογραφία.

Name	Status	Type	Order	Modified
8bit_adder.vhd	✓	VHDL	1	06/10/2020 07:44:15 ...
ALU.vhd	✓	VHDL	4	06/11/2020 09:51:24 ...
au.vhd	✓	VHDL	2	06/11/2020 08:15:39 ...
full_adder.vhd	✓	VHDL	0	06/10/2020 06:18:54 ...
LU.vhd	✓	VHDL	3	06/11/2020 09:46:17 ...

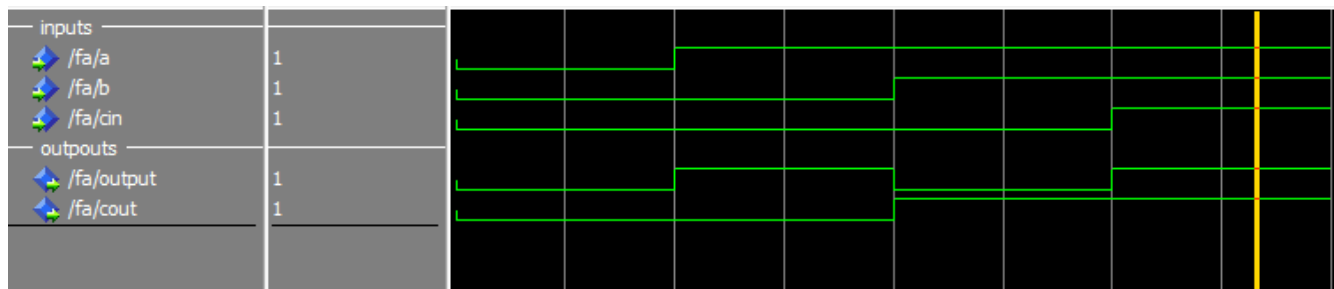
Επίπεδο πλήρους αθροιστή

```
library ieee;
use ieee.std_logic_1164.all;
entity fa is
port(a,b,cin:in std_logic;
      output,cout: out std_logic);
end fa;
architecture full_adder of fa is
begin
    output<=(a xor b) xor cin;
    cout<=((a xor b) and cin) or (a and b);
end full_adder;
```

Αυτό αποτελεί το χαμηλότερο επίπεδο σχεδίασης και είναι ουσιαστικά ένας πλήρης αθροιστής του 1 μπιτ με σήματα εισόδου 2 μπιτ και ένα κρατούμενο και 2 σήματα εξόδου του 1 μπιτ το αποτέλεσμα και το κρατούμενο .Υλοποιήθηκε με πύλες σύμφωνα με την λογική



ΑΠΟΤΕΛΕΣΜΑΤΑ



ΕΠΙΠΕΔΟ ΙΕΡΑΡΧΙΑΣ ΑΘΡΟΙΣΤΗΣ 8BIT

```

library ieee;
use ieee.std_logic_1164.all;
entity bit8_fa is
port(A,B:in std_logic_vector(7 downto 0);
      C: out std_logic_vector(7 downto 0);
      COUT:out std_logic);
end bit8_fa;
architecture bit8_arch of bit8_fa is

    signal c_out:std_logic_vector(6 downto 0);
    signal temp_carry:std_logic;
    signal temp_msb:std_logic;
    component fa
        port(a,b,cin:in std_logic;
              output,cout:out std_logic);
    end component ;
begin
    fa0:fa port map(A(0),B(0),'0',C(0),c_out(0));
    generate_label:
    for i in 1 to 6 generate
        fa_i:fa port map(A(i),B(i),c_out(i-1),C(i),c_out(i));
    end generate;
    fa7:fa port map(A(7),B(7),c_out(6),temp_msb,temp_carry);
    C(7)<=temp_msb;

    COUT<='1' when (temp_msb='1' and A(7)='0' and B(7)='0')else
    temp_carry;

end bit8_arch;

```

Αξίζει να σημειωθεί ότι έχουμε μια ειδική περίπτωση παραγωγής κρατούμενο αν το 8^ο μπιτ είναι 0 και οι 2 προσθετέοι είναι θετικοί διότι έχουμε αναπαράσταση σε συμπλήρωμα ως προς 2 και το αποτέλεσμα χρειάζεται άλλο ένα 0 για να μην είναι αρνητικό. Για την πρόσθεση μπιτ μπιτ χρησιμοποιήθηκε ο προηγούμενος κώδικας.

ΑΠΟΤΕΛΕΣΜΑΤΑ

	Msgs					
/bit8_fa/A	00000000	00000001	11111111	00000000		
/bit8_fa/B	00000000	01111111		00000000		
/bit8_fa/C	00000000	10000000	01111110	00000000		
/bit8_fa/COUT	0					

Επίπεδο AU(Arithmetic Unit)

```
library ieee;
use ieee.std_logic_1164.all;
use ieee.std_logic_unsigned.all;
entity Arethmetic_Unit is
    port (OpCode :in std_logic_vector (2 downto 0);
          A,B: in std_logic_vector (7 downto 0);
          C: out std_logic_vector (7 downto 0);
          Zero, Carry: out std_logic);
end Arethmetic_Unit;
architecture AU of Arethmetic_Unit is
    component bit8_fa is
        port(A,B:in std_logic_vector(7 downto 0);
             C: out std_logic_vector(7 downto 0);
             COUT:out std_logic);
    end component;
    signal B_Comp,B_temp: std_logic_vector (7 downto 0);
    signal C_add,C_sub: std_logic_vector (7 downto 0);
    signal temp_carry,CarryA,CarryS: std_logic ;
begin
    B_temp <= not B;
    B_Comp <= B_temp + 1;
    B_Add: bit8_fa port map (A, B, C_add, CarryA);
    B_Sub: bit8_fa port map (A, B_Comp, C_sub, CarryS);
```

```
C<=C_add(7)&C_add(6)&C_add(5)&C_add(4)&C_add(3)&C_add(2)&C_add(1)&C_add(0) when ((OpCode)="000") else
C_sub(7)&C_sub(6)&C_sub(5)&C_sub(4)&C_sub(3)&C_sub(2)&C_sub(1)&C_sub(0) when ((OpCode)="001" )else "00000000" ;

temp_carry<= '1' when (((A(7)='1')and (B(7)='1') and (C_add(7)='0' or C_sub(7)='0')) or ((A(7)='0') and (B(7)='0') and (C_add(7)='1' or C_sub(7)='1'))and (OpCode="000" or OpCode="001"
'0';
Carry<=temp_carry;

Zero<='1' when ((temp_carry='0' and (C_add)="00000000" and (OpCode="000"))or (temp_carry='0' and (C_sub)="00000000" and (OpCode="001"))) else
'0';
nd AU;
```

Αν είχαμε υπερχείλιση δεν επεξεργαζόμασταν κάπως το αποτέλεσμα απλά σηκώναμε το carry flag.

ΑΠΟΤΕΛΕΣΜΑΤΑ

	OpCode	A	B	C	Zero	Carry
/arethmetic_unit/OpCode	000	01100000	01000001	10100001	0	1
/arethmetic_unit/A	01100000	01100000	01000001	10100001	0	1
/arethmetic_unit/B	01000001	01100000	01000001	10100001	0	1
/arethmetic_unit/C	10100001	01100000	01000001	10100001	0	1
/arethmetic_unit/Zero	0	01100000	01000001	10100001	0	1
/arethmetic_unit/Carry	1	01100000	01000001	10100001	0	1

Επίπεδο LU(Logic Unit)

```

-- 11/06/2020
-- Logic Unit
library ieee;
    use ieee.std_logic_1164.all;
    use ieee.std_logic_unsigned.all;

    entity logic_unit is
        port (OpCode :in std_logic_vector (2 downto 0);
              A,B: in std_logic_vector (7 downto 0);
              C: out std_logic_vector (7 downto 0);
              zero: out std_logic);
    end logic_unit;
    architecture LU of logic_unit is
        begin
            C<=A and B when OpCode="100" else
            A or B when OpCode="110" else
            A xor B when OpCode="111" else
            (not A) when OpCode="101";
            zero<='1' when ((A xor B)="00000000" and OpCode="111" )or
            ((A or B)="00000000" and OpCode="110")or
            ((A and B)="00000000" and OpCode="100")or
            (A="11111111" and Opcode="101")else '0';
        end LU;
    end LU;

```

Να σημειωθεί ότι στη λογική μονάδα υπήρχε μόνο zero flag και όχι carry flag.

ΑΠΟΤΕΛΕΣΜΑΤΑ

Msgs									
/logic_unit/OpCode	100	100	101	110	111	100			
/logic_unit/A	00000000	01100111				00000000			
/logic_unit/B	10001101	10001101							
/logic_unit/C	00000000	00000101	10011000	11101111	11101010	00000000			
/logic_unit/zero	1								

Επίπεδο ALU(Arithmetic Logic Unit)

```

library ieee;
use ieee.std_logic_1164.all;

entity Arethmetic_Logic_Unit is
    port (Op :in std_logic_vector (2 downto 0);
          A,B: in std_logic_vector (7 downto 0);
          OUTPUT: out std_logic_vector (7 downto 0);
          Zero, Cout: out std_logic);
end Arethmetic_Logic_Unit;

architecture ALU of Arethmetic_Logic_Unit is
    component logic_unit
        port (OpCode :in std_logic_vector (2 downto 0);
              A,B: in std_logic_vector (7 downto 0);
              C: out std_logic_vector (7 downto 0);
              Zero: out std_logic);
    end component;
    Component Arethmetic_Unit
    port (OpCode :in std_logic_vector (2 downto 0);
          A,B: in std_logic_vector (7 downto 0);
          C: out std_logic_vector (7 downto 0);
          Zero, Carry: out std_logic);
    end component;
    signal output_temp1,output_temp2:std_logic_vector(7 downto 0);
begin
    AU1: Arethmetic_Unit port map (Op, A, B, output_temp1, Zero, Cout);
    LU1: logic_unit port map (Op, A, B, output_temp2, Zero);
    OUTPUT<=output_temp1 when Op(2)='0'else
        output_temp2;

end ;

```

ΑΠΟΤΕΛΕΣΜΑΤΑ

	110	000	001	100	101	110	111
/arethmetic_logic_unit/Op	110	000	001	100	101	110	111
/arethmetic_logic_unit/A	01010001	01010001	01010001	01010001	01010001	01010001	01010001
/arethmetic_logic_unit/B	00100111	01100111	00100111	01100111	01100111	01100111	01100111
/arethmetic_logic_unit/OUTPUT	01110111	10111000	00101010	00000001	10101110	01110111	01110110
/arethmetic_logic_unit/Zero	0						
/arethmetic_logic_unit/Cout	0						

ΑΣΚΗΣΗ 2

16-bit Carry-Select Adder

Για την άσκηση χρησιμοποιήθηκαν 7 files, στο 1^ο ιεραρχικό επίπεδο τα 3 αρχεία carry_select_adders, στο 2^ο ιεραρχικό επίπεδο τα αρχεία 2_bit_adders, 4_bit_adders, 8_bit_adders και στο τελευταίο ιεραρχικό επίπεδο το αρχείο full_adder.

Name	Status	Type	Order	Modified
2-bit-adder.vhd	✓	VHDL	3	06/12/2020 07:48:39 ..
carry_select_adde...	✓	VHDL	0	06/12/2020 06:18:28 ..
carry_select_adde...	✓	VHDL	4	06/12/2020 07:11:22 ..
carry_select_adde...	✓	VHDL	6	06/12/2020 07:58:33 ..
8-bit-adder.vhd	✓	VHDL	5	06/12/2020 07:35:19 ..
4-bit-adder.vhd	✓	VHDL	2	06/12/2020 05:18:13 ..
full_adder.vhd	✓	VHDL	1	06/12/2020 04:17:33 ..

a)

Carry select adder με block των 4 bits

Αρχικά, υλοποιήθηκε ο full adder, όπως παρακάτω:

```
library ieee;
use ieee.std_logic_1164.all;

entity full_adder is
    port(fa_in1,fa_in2,fa_cin : in STD_LOGIC;
         fa_out1,fa_cout : out STD_LOGIC
    );
end full_adder;

architecture full_adder_arch of full_adder is
begin
    fa_out1 <= fa_in1 xor fa_in2 xor fa_cin;
    fa_cout <= (fa_in1 and fa_in2) or (fa_cin and fa_in1) or (fa_cin and fa_in2);
end full_adder_arch;
```


Ο full adder έπειτα χρησιμοποιήθηκε για την υλοποίηση του 4 bit adder, ως component, κάνοντας χρήση της εντολής

```
library ieee;
use ieee.std_logic_1164.all;

entity four_bit_adder is
    port(fba_in1,fba_in2 : in STD_LOGIC_VECTOR (3 downto 0);
          fba_out1 : out STD_LOGIC_VECTOR (3 downto 0);
          fba_cin: in STD_LOGIC;
          fba_cout : out STD_LOGIC
    );
end four_bit_adder;

architecture four_bit_adder_arch of four_bit_adder is

    component full_adder
        port(fa_in1,fa_in2,fa_cin : in STD_LOGIC;
              fa_out1,fa_cout : out STD_LOGIC
        );
    end component full_adder;

    signal carries : STD_LOGIC_VECTOR (3 downto 0);

begin
    add1: full_adder port map( fa_in1 => fba_in1(0), fa_in2 => fba_in2(0),
                               fa_cin => fba_cin, fa_out1 => fba_out1(0), fa_cout => carries(0)) ;
    gen: for i in 1 to 3 generate
        add2: full_adder port map( fa_in1 => fba_in1(i), fa_in2 => fba_in2(i),
                                   fa_cin => carries(i-1), fa_out1 => fba_out1(i), fa_cout => carries(i)) ;
    end generate gen;
    fba_cout <= carries(3);

end four_bit_adder_arch;
```

Αντίστοιχα, χρησιμοποιώντας τον 4 bit adder ως component, δημιουργήθηκε ο carry select 16 bit adder με blocks των 4 bits.

```

library IEEE;
use IEEE.STD_LOGIC_1164.ALL;

entity carry_select_adder_4_block is
    Port ( in1, in2 : in STD_LOGIC_VECTOR (15 downto 0);
          out1 : out STD_LOGIC_VECTOR (15 downto 0);
          cout : out STD_LOGIC);
end carry_select_adder_4_block;

architecture carry_select_adder_4_block_arch of carry_select_adder_4_block is

    component four_bit_adder
        port(fba_in1 : in STD_LOGIC_VECTOR (3 downto 0);
             fba_in2 : in STD_LOGIC_VECTOR (3 downto 0);
             fba_out1 : out STD_LOGIC_VECTOR (3 downto 0);
             fba_cin: in STD_LOGIC;
             fba_cout : out STD_LOGIC
             );
    end component four_bit_adder;

    signal s : STD_LOGIC_VECTOR (21 downto 0);
    signal s1 : STD_LOGIC_VECTOR (15 downto 0);
    signal s2 : STD_LOGIC_VECTOR (15 downto 0);

begin
    m1: four_bit_adder port map( fba_in1 => in1(3 downto 0), fba_in2 => in2(3 downto 0),
                                fba_out1 => out1(3 downto 0), fba_cin => '0', fba_cout => s(0)) ;

    gen: for i in 1 to 3 generate
        m2: four_bit_adder port map( fba_in1 => in1((4*i+3) downto (4*i)), fba_in2 => in2((4*i+3) downto (4*i)),
                                    fba_out1 => s1((4*i+3) downto (4*i)), fba_cin => '0', fba_cout => s(3*i-2)) ;
        m3: four_bit_adder port map( fba_in1 => in1((4*i+3) downto (4*i)), fba_in2 => in2((4*i+3) downto (4*i)),
                                    fba_out1 => s2((4*i+3) downto (4*i)), fba_cin => '1', fba_cout => s(3*i-1)) ;
        out1((4*i+3) downto (4*i)) <= s1((4*i+3) downto (4*i)) when s(3*i-3) = '0' else s2((4*i+3) downto (4*i));
        s(3*i) <= (s(3*i-3) and s(3*i-1)) or s(3*i-2);
    end generate gen;

    cout <= s(9);

end carry_select_adder_4_block_arch;

```

Αποτελέσματα:

	Msgs									
/carry_select_adder_4_block	-1	16947	6408	0	-1					
/carry_select_adder_4_block	-1	6337	8724	0	-1					
/carry_select_adder_4_block	-2	23284	15132	0	-2					
/carry_select_adder_4_block	1									

Τα αποτελέσματα (σε δεκαδική μορφή) επαληθεύουν τη σωστή λειτουργία, όταν δεν υπάρχει overflow.

b)

Carry select adder με block των 2 bits

Αρχικά, δημιουργήθηκε το component 2 bit adder:

```
library ieee;
use ieee.std_logic_1164.all;

entity two_bit_adder is
    port(tba_in1,tba_in2 : in STD_LOGIC_VECTOR (1 downto 0);
          tba_out1 : out STD_LOGIC_VECTOR (1 downto 0);
          tba_cin: in STD_LOGIC;
          tba_cout : out STD_LOGIC
    );
end two_bit_adder;

architecture two_bit_adder_arch of two_bit_adder is

    component full_adder
        port(fa_in1,fa_in2,fa_cin : in STD_LOGIC;
              fa_out1,fa_cout : out STD_LOGIC
        );
    end component full_adder;

    signal carry: STD_LOGIC;

begin
    add1: full_adder port map( fa_in1 => tba_in1(0), fa_in2 => tba_in2(0),
                               fa_cin => tba_cin, fa_out1 => tba_out1(0), fa_cout => carry) ;
    add2: full_adder port map( fa_in1 => tba_in1(1), fa_in2 => tba_in2(1), fa_cin => carry,
                               fa_out1 => tba_out1(1), fa_cout => tba_cout) ;

end two_bit_adder_arch;
```

Έπειτα, ο αθροιστής με τα 8x2-bit blocks:

```
library IEEE;
use IEEE.STD_LOGIC_1164.ALL;

entity carry_select_adder_2_block is
    Port ( in1, in2 : in STD_LOGIC_VECTOR (15 downto 0);
           out1 : out STD_LOGIC_VECTOR (15 downto 0);
           cout : out STD_LOGIC);
end carry_select_adder_2_block;

architecture carry_select_adder_2_block_arch of carry_select_adder_2_block is

    component two_bit_adder
        port(tba_in1 : in STD_LOGIC_VECTOR (1 downto 0);
              tba_in2 : in STD_LOGIC_VECTOR (1 downto 0);
              tba_out1 : out STD_LOGIC_VECTOR (1 downto 0);
              tba_cin: in STD_LOGIC;
              tba_cout : out STD_LOGIC
        );
    end component two_bit_adder;

end carry_select_adder_2_block_arch;
```

```

signal s : STD_LOGIC_VECTOR (21 downto 0);
signal s1 : STD_LOGIC_VECTOR (15 downto 0);
signal s2 : STD_LOGIC_VECTOR (15 downto 0);

begin
  m1: two_bit_adder port map( tba_in1 => in1(1 downto 0), tba_in2 => in2(1 downto 0),
    tba_out1 => out1(1 downto 0), tba_cin => '0', tba_cout => s(0)) ;

  gen: for i in 1 to 7 generate
    m2: two_bit_adder port map( tba_in1 => in1((2*i+1) downto (2*i)), tba_in2 => in2((2*i+1) downto (2*i)),
      tba_out1 => s1((2*i+1) downto (2*i)), tba_cin => '0', tba_cout => s(3*i-2)) ;
    m3: two_bit_adder port map( tba_in1 => in1((2*i+1) downto (2*i)), tba_in2 => in2((2*i+1) downto (2*i)),
      tba_out1 => s2((2*i+1) downto (2*i)), tba_cin => '1', tba_cout => s(3*i-1)) ;
    out1((2*i+1) downto (2*i)) <= s1((2*i+1) downto (2*i)) when s(3*i-3)= '0' else s2((2*i+1) downto (2*i));
    s(3*i) <= (s(3*i-3) and s(3*i-1)) or s(3*i-2);
  end generate gen;

  cout <= s(21);

end carry_select_adder_2_block_arch;

```

Αποτελέσματα:

/carry_select_adder_2_block_arch	-1	-1	-12386	-17933	-30318	30317	17932	12385	0
/carry_select_adder_2_block_arch	-1	-1	31647	-12557	19091	-19092	12556	-31648	0
/carry_select_adder_2_block_arch	-2	-2	19261	-30490	-11227	11225	30488	-19263	0
/carry_select_adder_2_block_arch	1	1							

Τα αποτελέσματα (σε δεκαδική μορφή) επαληθεύουν τη σωστή λειτουργία, όταν δεν υπάρχει overflow.

Carry select adder με block των 8 bits

Αρχικά, δημιουργήθηκε το component 8 bit adder:

```

library ieee;
use ieee.std_logic_1164.all;

entity eight_bit_adder is
  port(eba_in1, eba_in2 : in STD_LOGIC_VECTOR (7 downto 0);
    eba_out1 : out STD_LOGIC_VECTOR (7 downto 0);
    eba_cin : in STD_LOGIC;
    eba_cout : out STD_LOGIC);
end eight_bit_adder;

```

```

architecture eight_bit_adder_arch of eight_bit_adder is

    component full_adder
        port(fa_in1,fa_in2,fa_cin : in STD_LOGIC;
             fa_out1,fa_cout : out STD_LOGIC
             );
    end component full_adder;

    signal carries : STD_LOGIC_VECTOR (7 downto 0);

begin
    add1: full_adder port map( fa_in1 => eba_in1(0), fa_in2 => eba_in2(0), fa_cin => eba_cin,
                              fa_out1 => eba_out1(0), fa_cout => carries(0)) ;
    gen: for i in 1 to 7 generate
        add2: full_adder port map( fa_in1 => eba_in1(i), fa_in2 => eba_in2(i),
                                   fa_cin => carries(i-1), fa_out1 => eba_out1(i), fa_cout => carries(i)) ;
    end generate gen;
    eba_cout <= carries(7);

end eight bit adder arch;

```

Έπειτα, ο αθροιστής με τα 2x8-bit blocks:

```

library IEEE;
use IEEE.STD_LOGIC_1164.ALL;

entity carry_select_adder_8_block is
    Port ( in1, in2 : in STD_LOGIC_VECTOR (15 downto 0);
          out1 : out STD_LOGIC_VECTOR (15 downto 0);
          cout : out STD_LOGIC);
end carry_select_adder_8_block;

architecture carry_select_adder_8_block_arch of carry_select_adder_8_block is

    component eight_bit_adder
        port(eba_in1, eba_in2 : in STD_LOGIC_VECTOR (7 downto 0);
             eba_out1 : out STD_LOGIC_VECTOR (7 downto 0);
             eba_cin: in STD_LOGIC;
             eba_cout : out STD_LOGIC
             );
    end component eight_bit_adder;

    signal s : STD_LOGIC_VECTOR (2 downto 0);
    signal s1 : STD_LOGIC_VECTOR (15 downto 0);
    signal s2 : STD_LOGIC_VECTOR (15 downto 0);

begin
    m1: eight_bit_adder port map( eba_in1 => in1(7 downto 0), eba_in2 => in2(7 downto 0),
                                   eba_out1 => out1(7 downto 0), eba_cin => '0', eba_cout => s(0)) ;

    m2: eight_bit_adder port map( eba_in1 => in1(15 downto 8), eba_in2 => in2(15 downto 8),
                                   eba_out1 => s1(15 downto 8), eba_cin => '0', eba_cout => s(1)) ;
    m3: eight_bit_adder port map( eba_in1 => in1(15 downto 8), eba_in2 => in2(15 downto 8),
                                   eba_out1 => s2(15 downto 8), eba_cin => '1', eba_cout => s(2)) ;
    out1(15 downto 8) <= s1(15 downto 8) when s(0)='0' else s2(15 downto 8);

    cout <= (s(0) and s(2)) or s(1);

end carry_select_adder_8_block_arch;

```

Αποτελέσματα:

/carry_select_adde...	-1	-24158	24157	6213	17944	0				
/carry_select_adde...	-5	-23287	23282	6294	16996	4	0			
/carry_select_adde...	-6	18091	-18097	12507	-30596	4	0			
/carry_select_adde...	1									

Τα αποτελέσματα (σε δεκαδική μορφή) επαληθεύουν τη σωστή λειτουργία, όταν δεν υπάρχει overflow.