Abstract

This report describes the design and evalation of a 32-bit Carry Select Adder (CSA), evaluating how it compares to the conventional Carry Ripple Adder (CRA) in timing, complexity and implementation area.

1 Circuits and Coding

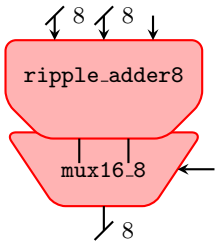


FIGURE 1: Diagram of carry ripple adder.

FIGURE 2: Diagram of carry select adder.

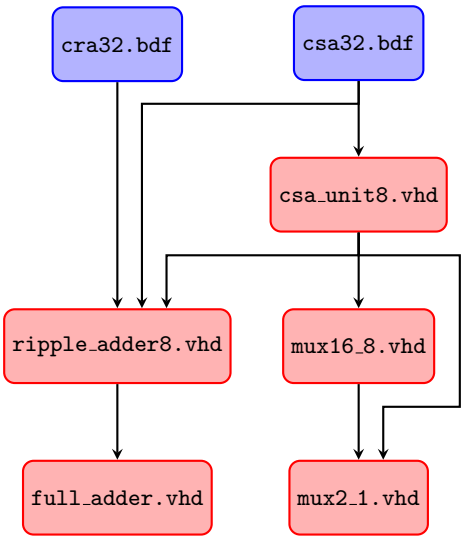


FIGURE 3: Hierarchy of entity declarations.

```
1 -- architecture
2 ARCHITECTURE full_adder_architecture OF full_adder IS
```

```
BEGIN
    Cout <= ((A xor B) and Cin) or (A and B);
    S <= (A xor B) xor Cin;
END;
```

Listing 1: `full_adder_architecture` declaration in `full_adder.vhd`

```
-- architecture
ARCHITECTURE mux2_1_architecture OF mux2_1 IS
BEGIN
    Q <= (A and not SEL) or (B and SEL);
END mux2_1_architecture;
```

Listing 2: `mux2_1_architecture` declaration in `mux2_1.vhd`

```
    -- map 8 muxes with common select S
    mux0 : mux2_1 port map (A => A(0),
                            B => B(0),
                            SEL => SEL,
                            Q => Q(0));
```

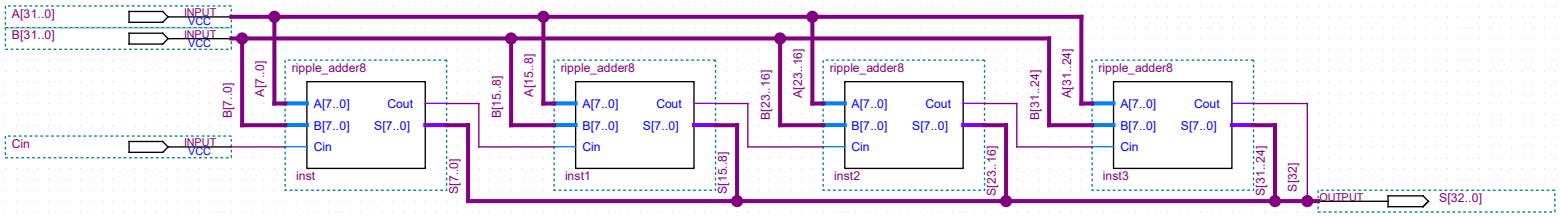
Listing 3: Snippet showing one of eight parallel `mux2_1` components mapped in `mux16_8.vhd` architecture declaration.

```
    SIGNAL C : std_logic_vector(6 downto 0);
BEGIN
    -- chain 8 full adders using intermediate carry signals
    fa0 : full_adder PORT MAP ( A => A(0),
                                B => B(0),
                                Cin => Cin,
                                Cout => C(0),
                                S => S(0));
```

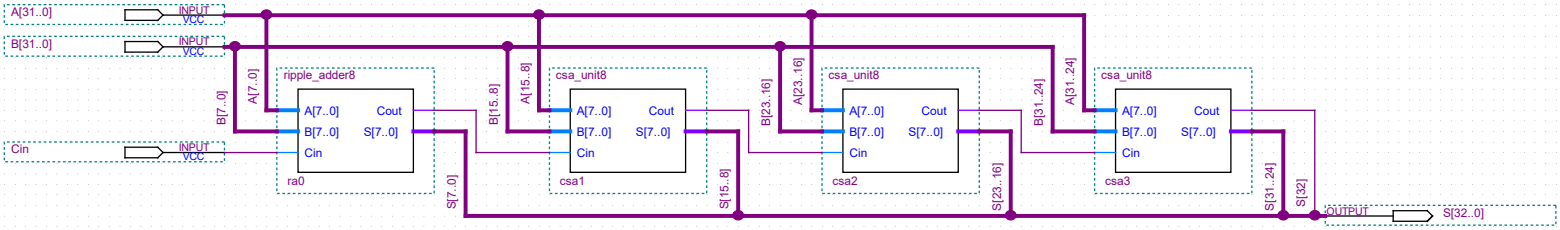
Listing 4: Snippet showing one of eight chained `full_adder` components mapped in `ripple_adder8.vhd` architecture declaration.

```
    -- candidates calculated by each ripple adder
    SIGNAL S0 : std_logic_vector(7 downto 0);
    SIGNAL Cout0 : std_logic;
    SIGNAL S1 : std_logic_vector(7 downto 0);
    SIGNAL Cout1 : std_logic;
BEGIN
    ra0 : ripple_adder8 PORT MAP (  A => A,
                                    B => B,
                                    Cin => '0',
                                    Cout => Cout0,
                                    S => S0);
    ra1 : ripple_adder8 PORT MAP (  A => A,
                                    B => B,
                                    Cin => '1',
                                    Cout => Cout1,
                                    S => S1);
    smux : mux16_8 PORT MAP (A => S0,
                             B => S1,
                             SEL => Cin,
                             Q => S);
    cmux : mux2_1 PORT MAP ( A => Cout0,
                             B => Cout1,
                             SEL => Cin,
                             Q => Cout);
```

Listing 5: Snippet showing how the two ripple adders were mapped to the intermediate candidate signals before being selected using multiplexors and the carry in port in `csa_unit8.vhd` architecture declaration.



(A) `cra32.bdf`: 32-bit carry ripple adder



(B) `csa32.bdf`: 32-bit carry select adder, 8-bit select units

FIGURE 4: Block diagram schematics for the top-level entities of each project.

2 Speed and Operation

3 FPGA Implementation