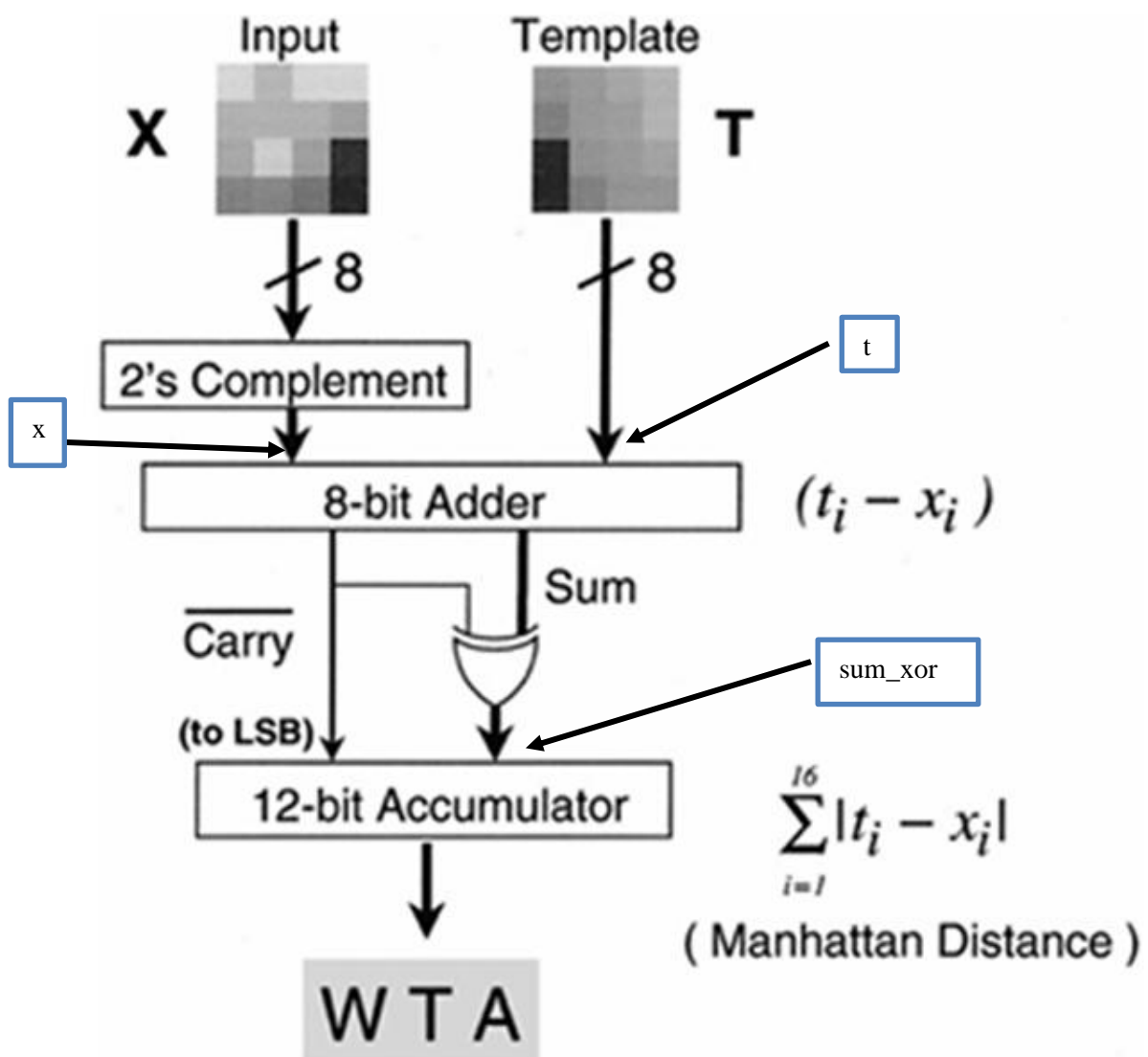


BÀI TẬP VỀ NHÀ TUẦN 9

Xử lý ảnh và video

Đề: Viết code verilog tính giá trị tuyệt đối và tích lũy giữa 2 vector dựa trên Fig 1 file ME_Hardware1.pdf.

- Đầu tiên, từ hình ảnh Fig 1 trong file, ta gọi giá trị sau một số bước như dưới



- Xây dựng mã Verilog cho thực hiện tính toán như sau

```

module Manhattan (Clk, RstN, X, T, WTA);
    input Clk;
    input RstN;
    input [7:0] X;
    input [7:0] T;
    output [11:0] WTA;

    reg signed [7:0] x,t;
    reg [11:0] WTA, tmp;
    reg [4:0] cnt = 1'b0;
    reg [8:0] sum, sum_xor;
    reg Ca;

always @(posedge Clk, negedge RstN) begin
    if(~RstN) begin
        WTA <= 12'b0;
        sum <= 1'b0;
        tmp <= 1'b0;
    end else begin
        x = ~X + 1'b1 ;
        t = T;
        sum = {1'b0,t} + {1'b0,x};
        Ca = sum[8] ;
        if (cnt == 16) begin
            WTA <= tmp;
            cnt <= 1'b0;
            tmp <= 1'b0;
        end else begin
            cnt <= cnt + 1'b1;
            if(Ca == 0) begin
                sum_xor = sum[7:0] ^ {~Ca,~Ca,~Ca,~Ca,~Ca,~Ca,~Ca,~Ca} ;
                tmp = tmp + sum_xor[7:0] + 1'b1;
            end else
                tmp <= tmp + sum[7:0] ;
        end
    end
end
endmodule

```

Tính x là số bù 2 của X

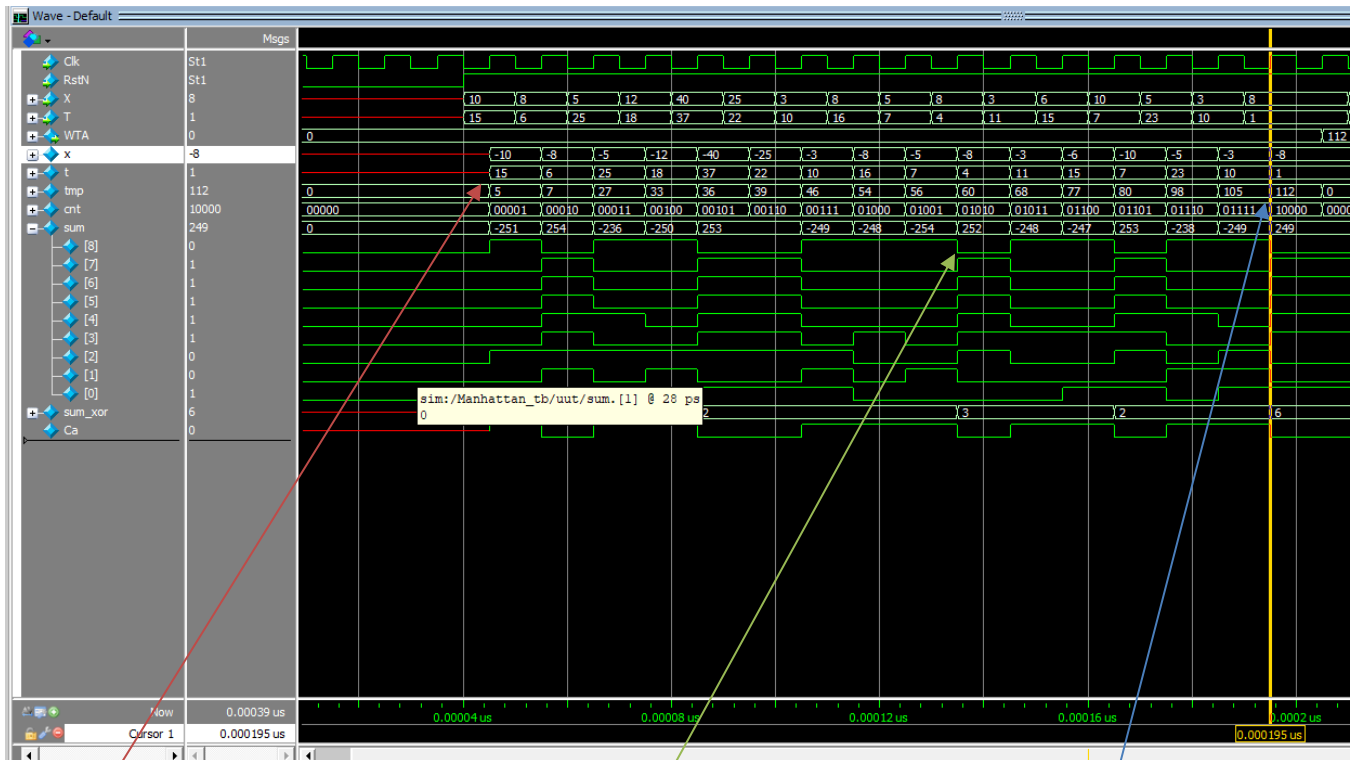
Mở rộng thêm bit 0 về MSB cho x và t để lấy bit Carry

Bit nhớ là, hay sum là số âm

Xor với bit carry (1) để đảo bit, rồi cộng thêm 1 để cộng dồn giá trị bù 2 của giá trị sum âm

- Kết quả mô phỏng testbench:

Dù đã tính ra đúng kết quả chính xác sau 16 clk kể từ lúc đọc được giá trị và để bắt đầu tính toán, nhưng mất thêm 1 clock nữa để đưa giá trị ra ngoài. Có thể khắc phục bằng việc đọc thanh ghi từ bên ngoài thay vì chờ mạch chốt kết quả cuối cùng.



Thời điểm bắt đầu
đọc được giá trị vào

Kết quả khi tính là âm
($-8 + 4 = -4$) nên bit
Carry (bit8) là 1, 8 bit
còn lại hiển thị đúng -4

Tính ra kết quả
sau 16 clock