Chương 3

ỨNG DỤNG FPGA TRONG THIẾT KẾ CÁC MODULE ĐIỀU KHIỂN

I. MỘT SỐ LƯU Ý KHI THIẾT KẾ

1.1. Các biến chỉ được gán trong 1 khối *always*, không được gán trong nhiều khối *always* khác

```
reg [7:0] a, b;
initial a = 4;

always @(posedge clk) begin
    a = b + 2;
end

always @(posedge reset) begin
    a = 0;
end
```

```
reg [7:0] a, b;
initial a = 4;

always @(posedge clk or posedge reset)
begin
    if (reset == 1)
        a = 0;
    else
        a = b + 2;
end
```

I. MỘT SỐ LƯU Ý KHI THIẾT KẾ

1.2. Để phát hiện xung cạnh lên của các tín hiệu, dùng xung clk tần số cao chèn vào

```
module demxung_encoder (clk, enc, D);
input clk, enc;
output [7:0] D;
reg [7:0] D = 8'h00;
reg pre_enc = 0;

always @(posedge clk) begin
    pre_enc <= enc;
    if ({pre_enc, enc} == 2'b01)
        D <= D + 1;
end
```

```
module demxung_encoder (clk, enc, D);
input clk, enc;
output [7:0] D;
reg [7:0] D = 8'h00;
reg pre_enc = 0;

always @(posedge clk) begin
    pre_enc <= enc;
end

always @(posedge clk) begin
    if ({pre_enc, enc} == 2'b01)
        D = D + 1;
end
```

I. MỘT SỐ LƯU Ý KHI THIẾT KẾ

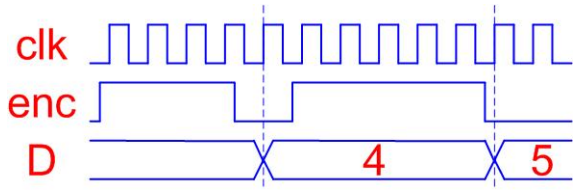
1.3. Dùng biến tạm để cập nhật giá trị các bộ đếm

```
module do_dorongxung (clk, enc, D);
input clk, enc;
output [7:0] D;
reg [7:0] D = 8'h00;
reg pre_enc = 0;

always @(posedge clk) begin
    pre_enc <= enc;
    if  ({pre_enc, enc} == 2'b01)
            D <= 1;
    else if  ({pre_enc, enc} == 2'b11)
            D <= D+1;
end
```

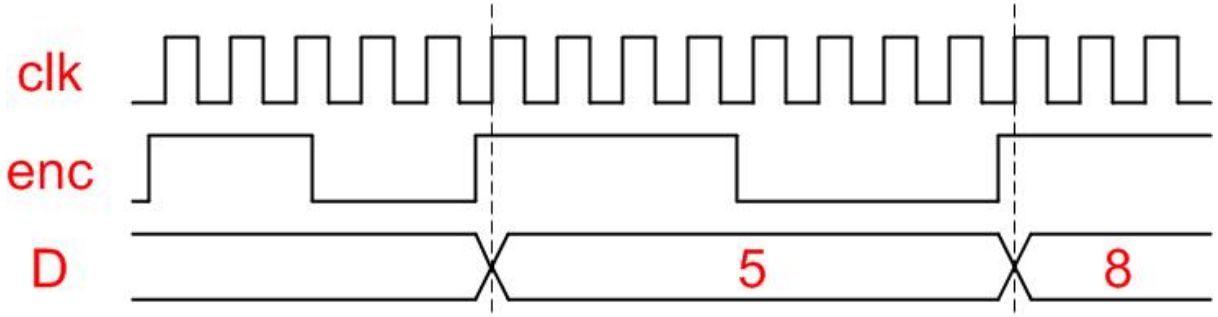
```
module do_dorongxung (clk, enc, D);
input clk, enc;
output [7:0] D;
reg [7:0] D = 8'h00, temp = 8'h01;
reg pre_enc = 0;

always @(posedge clk) begin
    pre_enc <= enc;
    if  ({pre_enc, enc} == 2'b11)
            temp <= temp+1;
    else if  ({pre_enc, enc} == 2'b10)  begin
            D <= temp; temp <= 0;
    end
end
```



I. MỘT SỐ LƯU Ý KHI THIẾT KẾ

1.4. Ví dụ 1: Viết chương trình verilog đo tần số



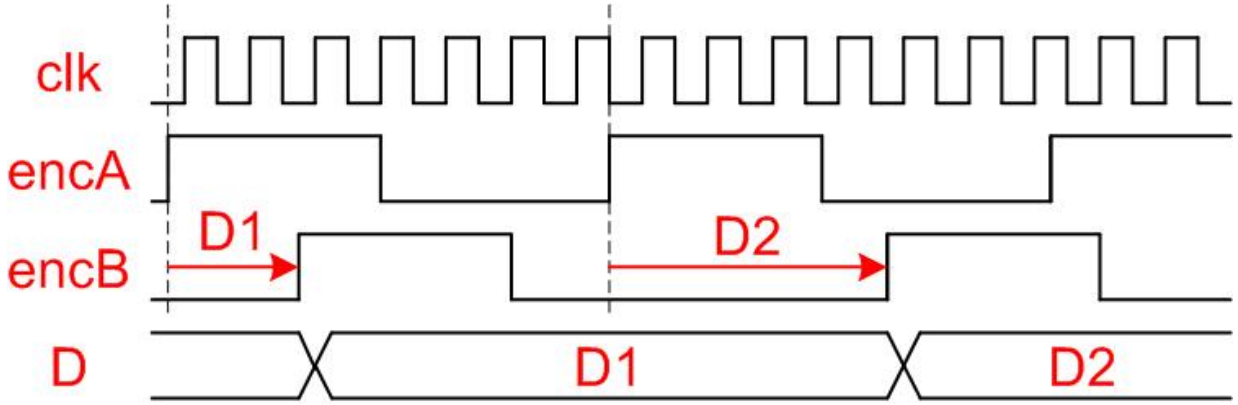
- Giá trị D được cập nhật sau mỗi cạnh lên của xung **enc.**
- Tần số xung **clk** chọn rất lớn so với tần số của xung **enc**
- Nếu bộ đếm D vượt quá giá trị 0xFF thì sẽ luôn bằng 0xFF

```
module do_tanso (clk, enc, D);
input clk, enc;
output [7:0] D;
reg [7:0] D = 8'h00, temp = 8'h00;
reg pre_enc = 0;

always @(posedge clk) begin
    …
end
```

I. MỘT SỐ LƯU Ý KHI THIẾT KẾ

1.4. Ví dụ 2: Viết chương trình verilog đo độ lệch pha 2 tín hiệu



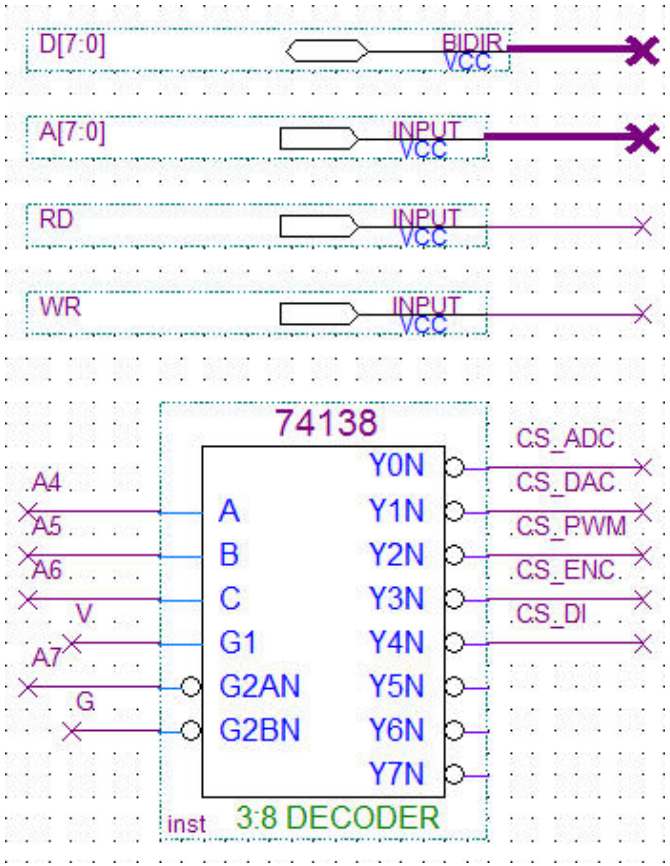
- Giá trị D được cập nhật sau mỗi cạnh lên của xung **encB.**
- Nếu bộ đếm D vượt quá giá trị 0xFF thì sẽ luôn bằng 0xFF
- Giả sử **encA** luôn luôn nhanh pha hơn **encB**

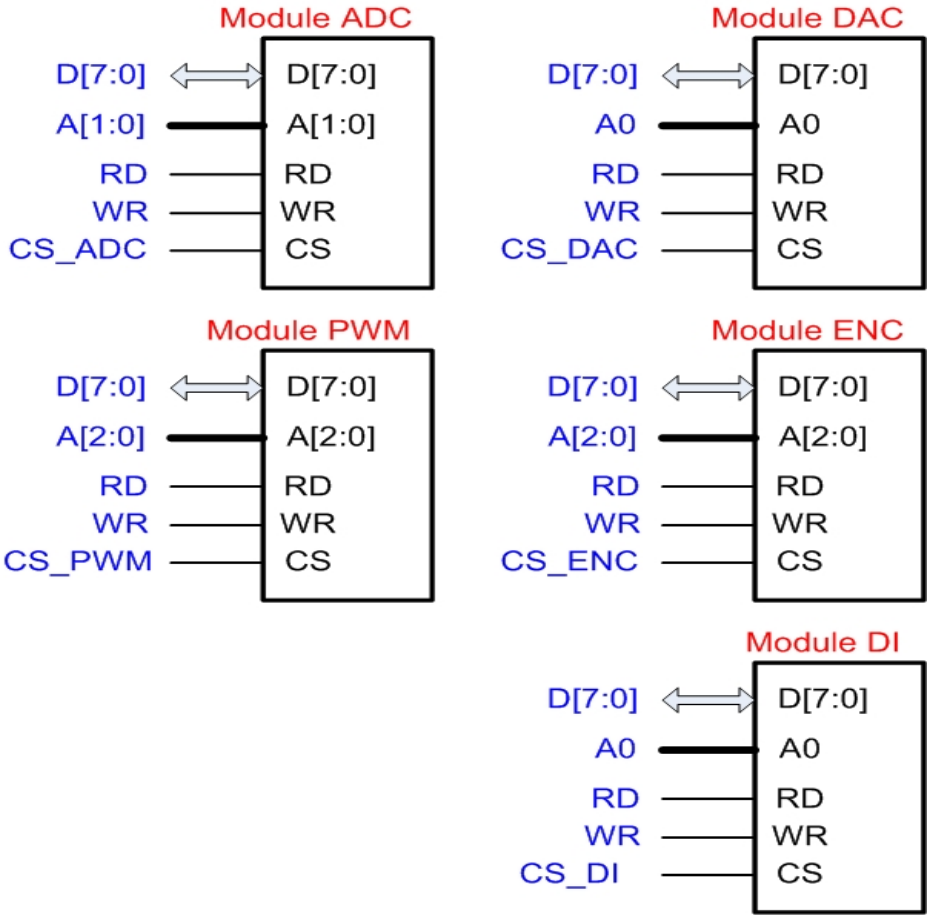
```
module do_dolechpha (clk, encA, encB, D);
input clk, encA, encB;
output [7:0] D;
reg [7:0] D = 8'h00, temp = 8'h00;
reg pre_enc = 0;

…
```

II. THIẾT KẾ CÁC MODULE ĐIỀU KHIỂN

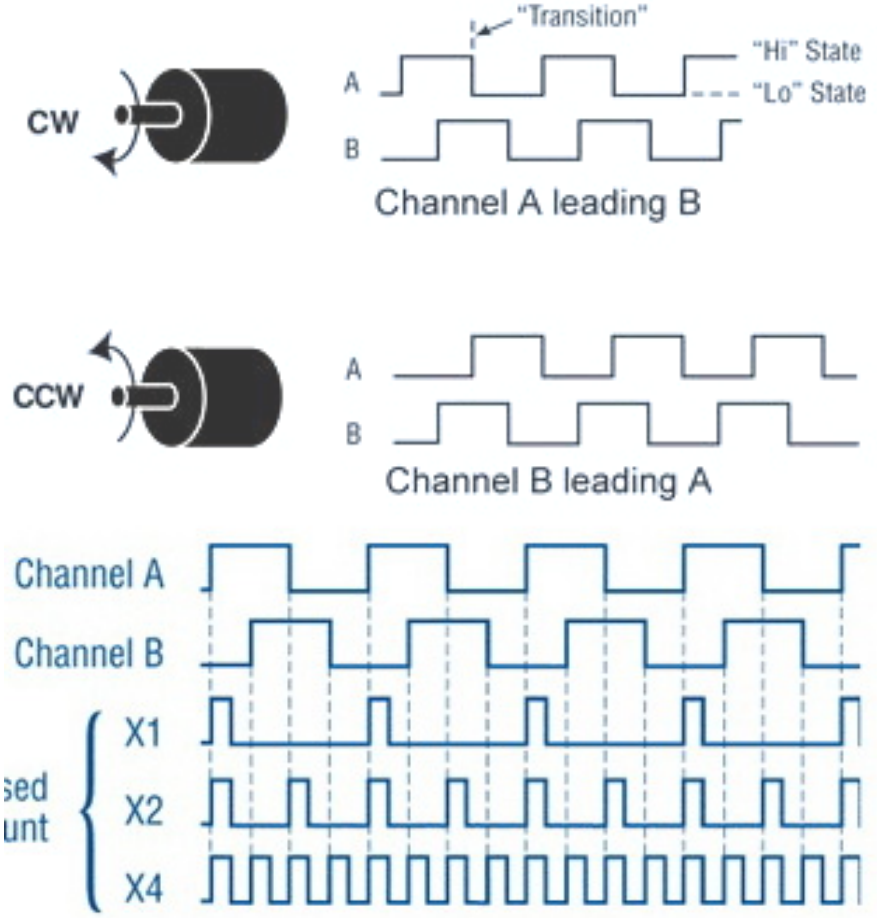
Sơ đồ kết nối phần cứng các module điều khiển





II. THIẾT KẾ CÁC MODULE ĐIỀU KHIỂN

1. Module đọc xung encoder



II. THIẾT KẾ CÁC MODULE ĐIỀU KHIỂN

1. Module đọc xung encoder

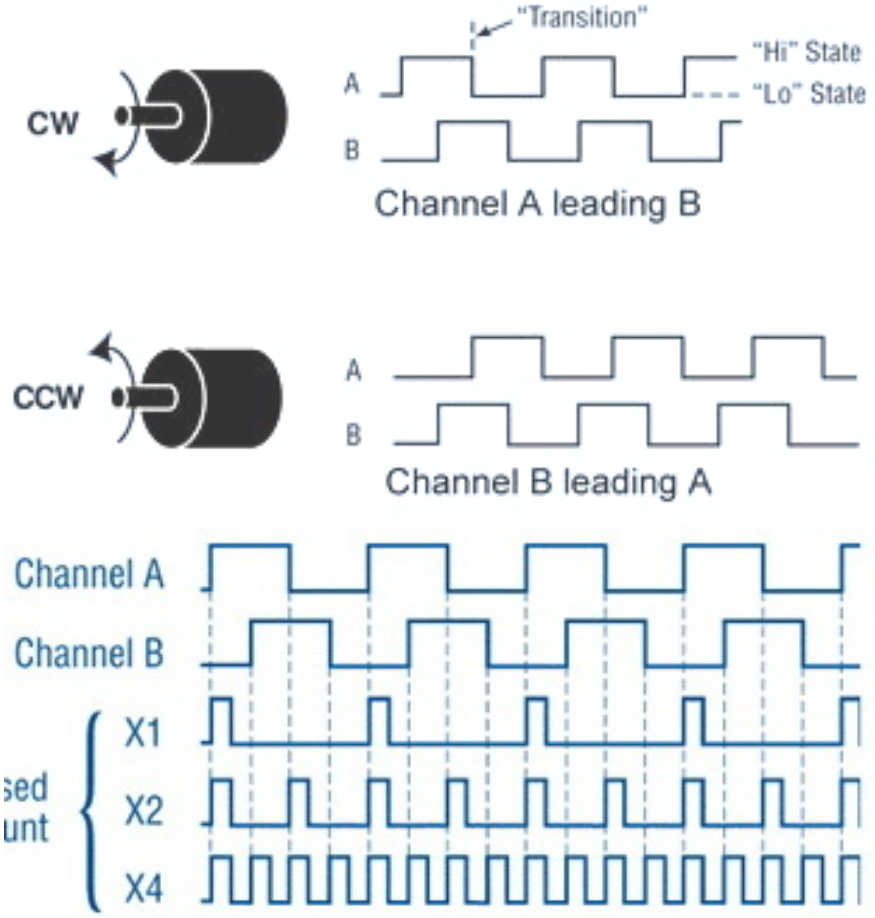
- Đầu vào clk, encA, encB, rst, x4
- Đầu ra 16 bit D[15:0]
- Khi rst = 1: D[15:0] = 16'h8000
- Khi rst = 0: D[15:0] đếm lên,

xuống theo xung encA, encB

- Đếm x1, x2, x4

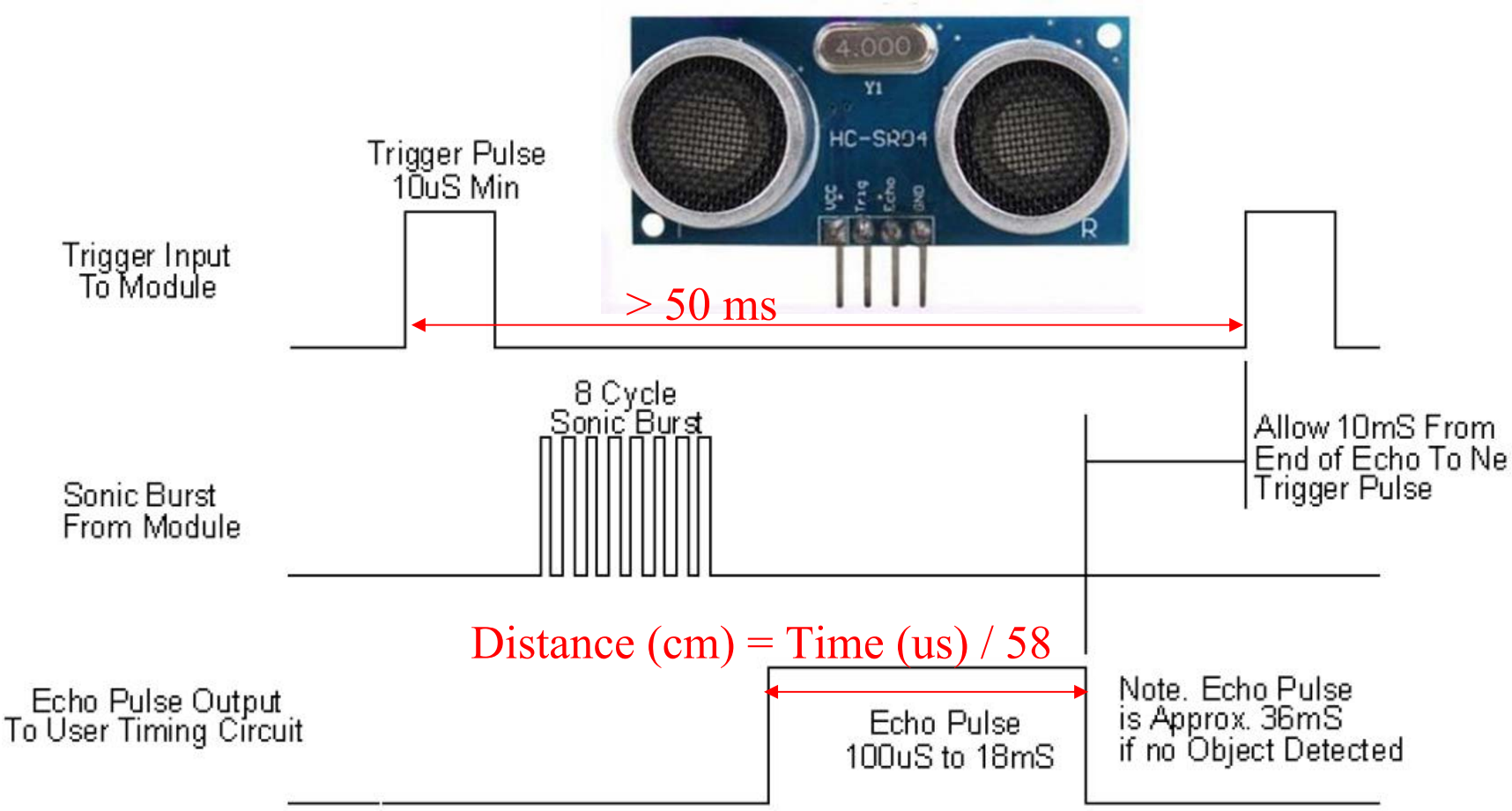
```
module encoder (clk,encA,encB,x4,rst,D);
input clk, encA, encB, rst;
input [1:0] x4;
output [15:0] D;
reg [15:0] D = 16'h8000;

…
```



II. THIẾT KẾ CÁC MODULE ĐIỀU KHIỂN

2. Module đọc cảm biến siêu âm

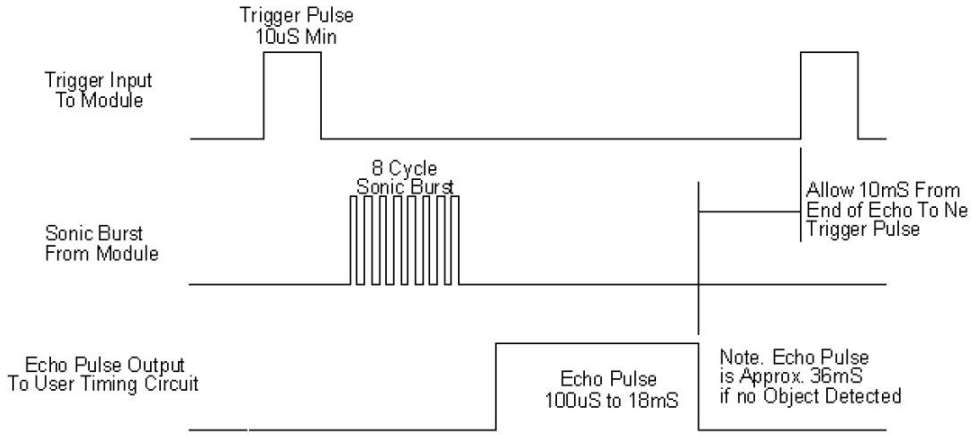


II. THIẾT KẾ CÁC MODULE ĐIỀU KHIỂN

2. Module đọc cảm biến siêu âm

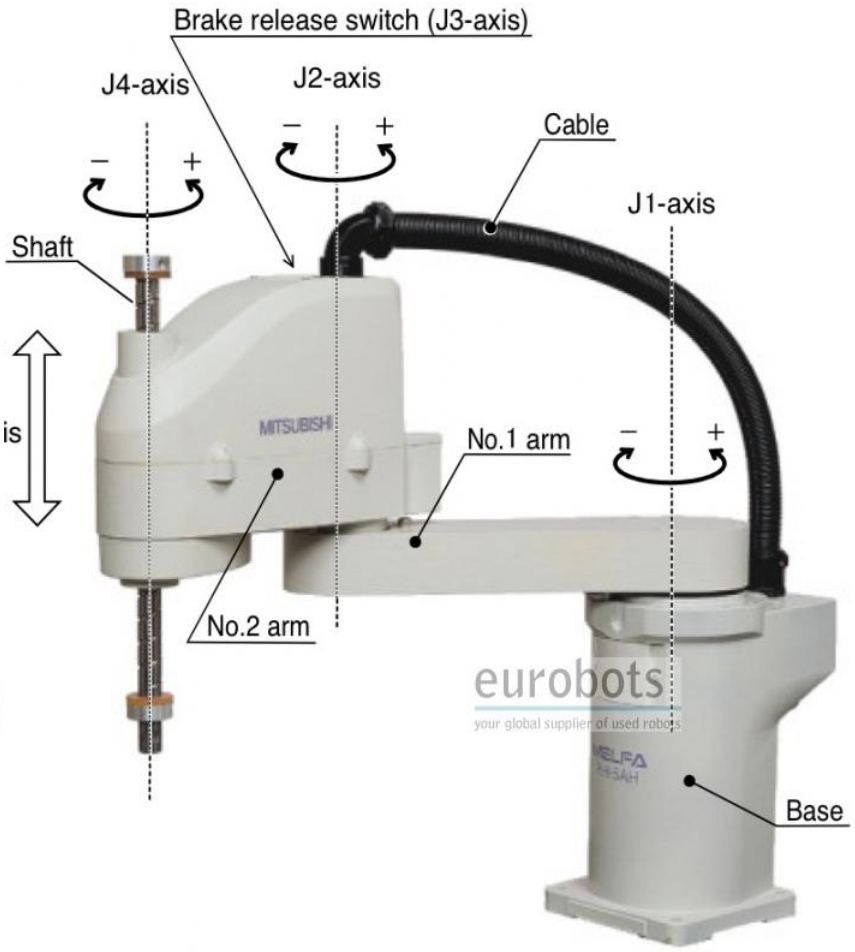
- Đầu vào clk (1us), EchoPulse
- Đầu ra 16 bit D[15:0], Trigger
- Chân Trigger có độ rộng xung 10us, lặp lại với chu kỳ T = 50ms (Chỉnh T = 1ms khi mô phỏng)
- Ngõ ra D[15:0] cập nhật khi có cạnh xuống của EchoPulse. (Đơn vị đo us).
- Nếu D = 0xFFFF thì không được tăng D.

```
module sfr04 (clk, EchoPusle, Trigger, D);
input clk, EchoPuse;
output Trigger;
output [15:0] D;
reg [15:0] D = 16'h0000;
…
```



II. THIẾT KẾ CÁC MODULE ĐIỀU KHIỂN

3. Module nội suy cho máy CNC, robot

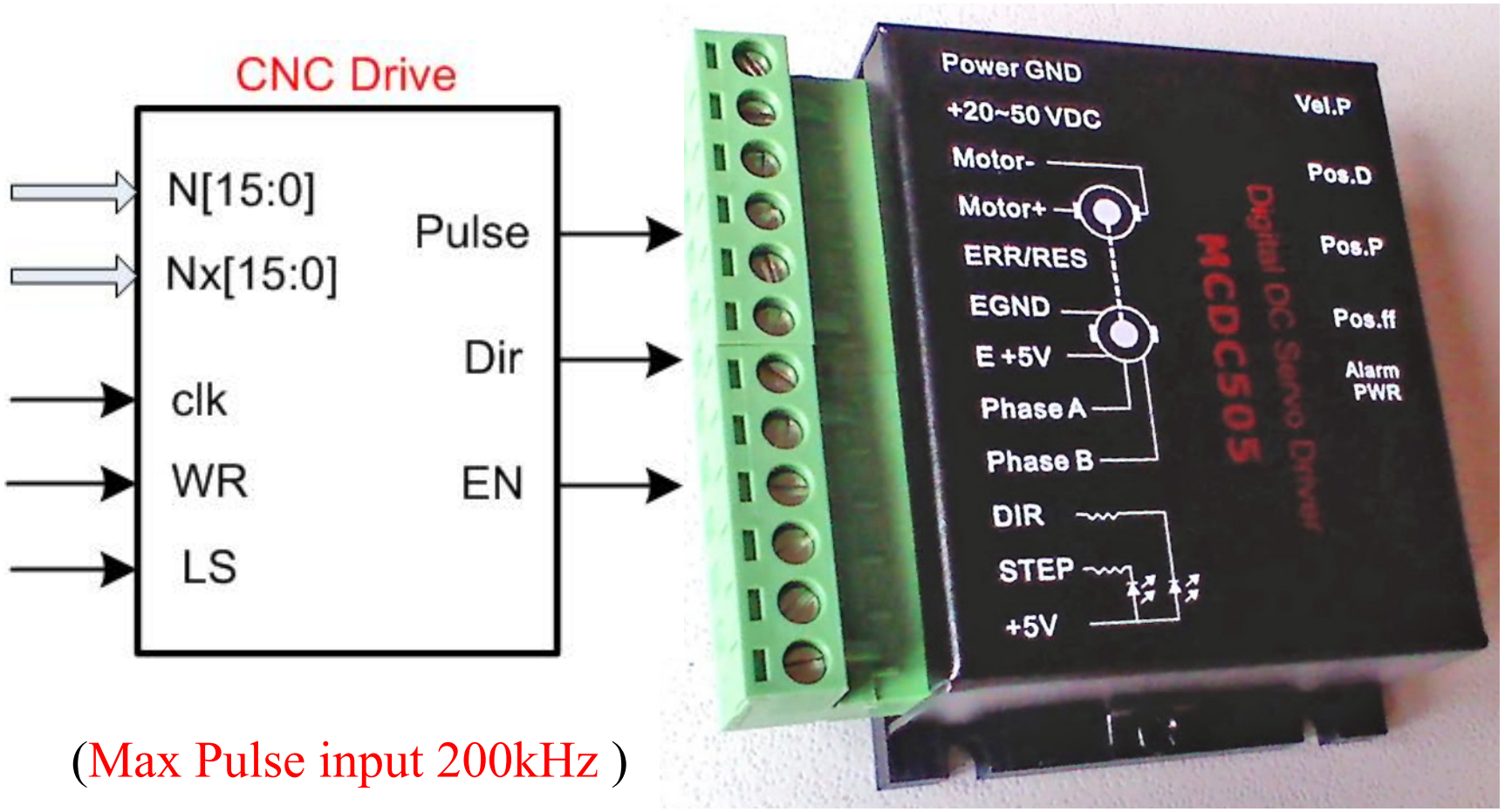


II. THIẾT KẾ CÁC MODULE ĐIỀU KHIỂN

3. Module nội suy cho máy CNC, robot

II. THIẾT KẾ CÁC MODULE ĐIỀU KHIỂN

3. Module nội suy cho máy CNC, robot

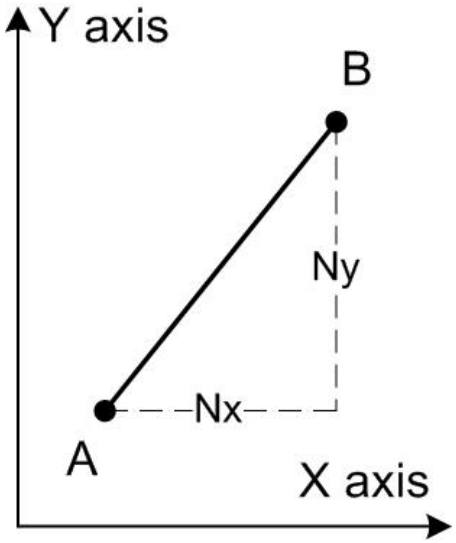


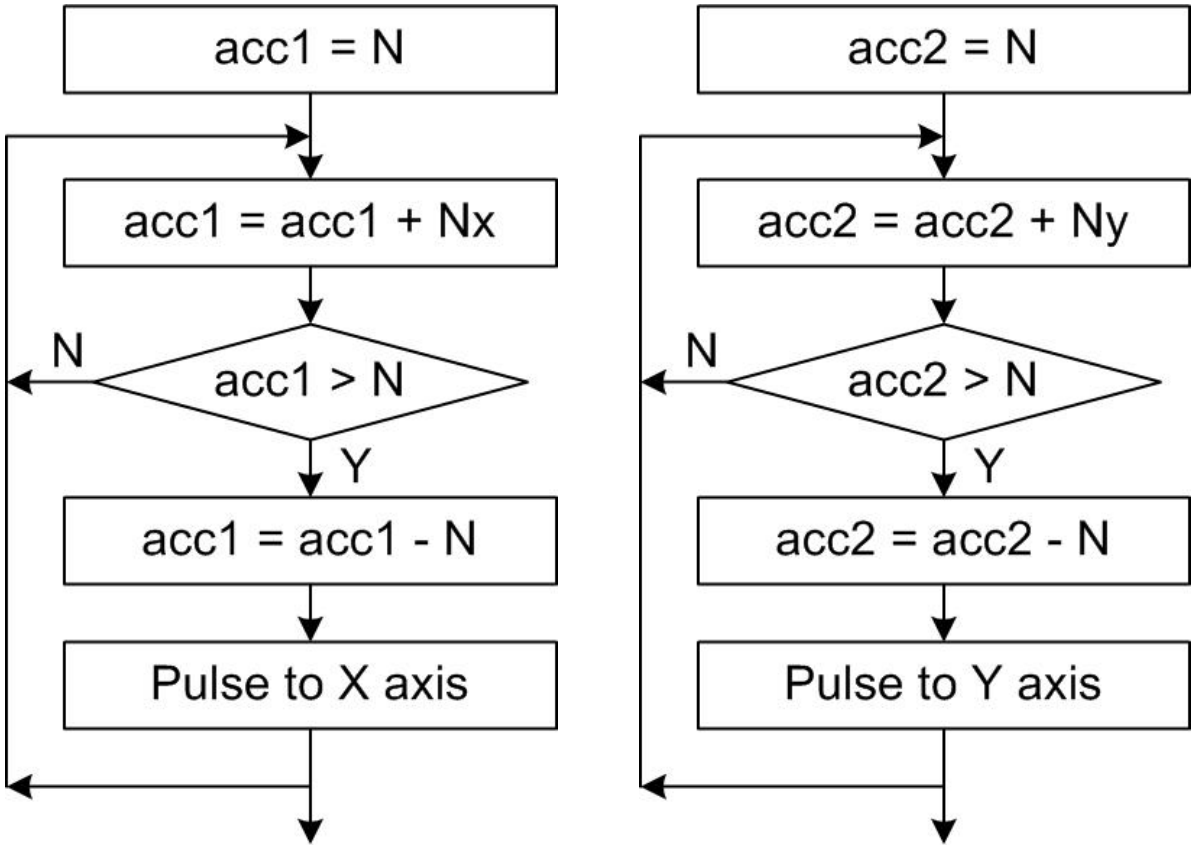
(Max Pulse input 200kHz)

II. THIẾT KẾ CÁC MODULE ĐIỀU KHIỂN

3. Module nội suy cho máy CNC, robot

- Thuật toán nội suy đường thẳng theo phương pháp xung chuẩn





- Nx: số xung trục X
- Ny: số xung trục Y
- N: số xung cực đại

II. THIẾT KẾ CÁC MODULE ĐIỀU KHIỂN

3. Module nội suy cho máy CNC, robot

- Đầu vào clk (1us), WR, LS
- Đầu vào N[7:0], Nx[7:0]
- Đầu ra Pulse, Dir
- Khi có xung cạnh lên của WR, Nx sẽ nạp giá trị mới.
- Khi LS = 1, ngõ ra Pulse = 0, LS = 0, Pulse xuất ra theo nội suy.
- Nx[7] qui định bit dấu cho Dir,
  Nx[7] = 1, Dir = 1.
  Nx [7] = 0, Dir = 0.
  Nx[6:0] là giá trị số xung cần xuất

```
module servo (clk,WR,LS, Nx,N, Pulse,Dir);
input  clk, WR, LS;
input [7:0]  N, Nx;
output Pulse, Dir;
…
always @(posedge clk) begin
     acc = acc + Nx;
     if (acc > N) begin
          acc = acc - N;   pinout = 1;
     end
     else    pinout = 0;
end
…
assign  Pulse = ~clk & pinout;
```