



اعضای گروه :

مهیار قاسمی

سید مصطفی عرب بنی هاشم

سید حسین بابایی

نحوه اجرای کد ها به صورت قسمت های جدا از هم با هدف ساده سازی برنامه  
برای شما انجام شده

شکل ۵\_۶ از کتاب معماری کامپیوتر موریس مانو به نام واحد های کنترل کامپیوتر است این شکل ساختار واحد کنترل را نشان می دهد که برای اجرای دستورالعمل در یک کامپیوتر استفاده می شود . در اینجا توضیح اجزای مختلف این شکل آمده است

- ثبات دستورالعمل (Instruction Register\_IR) :

ثبات دستورالعمل (IR) شامل ۱۶ بیت ، که بیت های ۰ تا ۱۱ مربوط به آدرس، بیت های ۱۱ تا ۱۴ مربوط به کد عملیات (opcode) و بیت ۱۵ جهت مشخص کردن مستقیم یا غیر مستقیم بودن آدرس (i) می باشد.

```
1 module InstructionRegister(  
2     input [15:0] instruction,  
3     input clk,  
4     output reg [15:0] IR  
5 );  
6     always @(posedge clk) begin  
7         IR <= instruction;  
8     end  
9 endmodule
```

- دیکدر ۳ به ۸ :

این دیکدر کد عملیات (opcode) را به سیگنال های کنترلی مربوط تبدیل می کند . ورودی های آن شامل بیت های ۱۲ تا ۱۴ ثابت دستورالعمل (IR) است و خروجی های آن شامل سیگنال های D0 تا D7 می باشد.

```
1 module Decoder3to8(  
2     input  [2:0] in,  
3     output reg [7:0] out  
4 );  
5     always @(*) begin  
6         case(in)  
7             3'b000: out = 8'b00000001;  
8             3'b001: out = 8'b00000010;  
9             3'b010: out = 8'b00000100;  
10            3'b011: out = 8'b00001000;  
11            3'b100: out = 8'b00010000;  
12            3'b101: out = 8'b00100000;  
13            3'b110: out = 8'b01000000;  
14            3'b111: out = 8'b10000000;  
15            default: out = 8'b00000000;  
16        endcase  
17    end  
18 endmodule  
19
```

- مدار کنترل :

مدار کنترل شامل گیت های منطقی و سیگنال های کنترلی است که برای اجرای دستورالعمل استفاده می شود . این مدار سیگنال های ورودی را از دیکدر و سایر ورودی ها دریافت می کند و سیگنال های کنترلی مناسب را برای اجزای مختلف کامپیوتر تولید می کند.

```

1 module ControlUnit(
2     input wire [7:0] D,
3     input wire [15:0] T,
4     input wire [15:0] IR,
5     output reg [3:0] control_signals
6 );
7     always @(*) begin
8
9         control_signals = 4'b0000;
10
11         if (T[0])
12             control_signals[0] = 1'b1;
13
14         if (T[1] && IR[11:10] == 2'b01)
15             control_signals[1] = 1'b1;
16
17         if (T[2] && IR[9:8] == 2'b10)
18             control_signals[2] = 1'b1;
19
20         if (T[3])
21             control_signals[3] = 1'b1;
22     end
23 endmodule

```

• توالی شمار (sequence counter \_sc) :

توالی شمار یک شمارنده چهار بیتی است که برای تولید توالی زمانی (T0 تا T15) استفاده می شود

```

1 module SequenceCounter(
2     input clk,
3     input clr,
4     input incr,
5     output reg [3:0] SC
6 );
7     always @(posedge clk or posedge clr) begin
8         if (clr)
9             SC <= 4'b0000;
10        else if (incr)
11            SC <= SC + 1;
12    end
13 endmodule

```

- دیکدر ۴ به ۱۶ :

این دیکدر سیگنال های توالی شمار ((sc را به توالی زمانی (T0 تا T15) تبدیل می کند که برای کنترل عملیات در مراحل مختلف اجرای دستورالعمل استفاده می شود

```

1  module Decoder4to16(
2      input [3:0] in,
3      output reg [15:0] out
4  );
5      always @(*) begin
6          case(in)
7              4'b0000: out = 16'b0000000000000001;
8              4'b0001: out = 16'b0000000000000010;
9              4'b0010: out = 16'b0000000000000100;
10             4'b0011: out = 16'b0000000000001000;
11             4'b0100: out = 16'b0000000000010000;
12             4'b0101: out = 16'b0000000000100000;
13             4'b0110: out = 16'b0000000001000000;
14             4'b0111: out = 16'b0000000010000000;
15             4'b1000: out = 16'b0000000100000000;
16             4'b1001: out = 16'b0000001000000000;
17             4'b1010: out = 16'b0000010000000000;
18             4'b1011: out = 16'b0000100000000000;
19             4'b1100: out = 16'b0001000000000000;
20             4'b1101: out = 16'b0010000000000000;
21             4'b1110: out = 16'b0100000000000000;
22             4'b1111: out = 16'b1000000000000000;
23             default: out = 16'b0000000000000000;
24         endcase
25     end
26 endmodule

```

- توضیح عملکرد کلی

ثبات دستورالعمل (IR):

دستورالعمل فعلی را نگه می دارد و کد عملیات (opcode) را به دیکدر ۳ به ۸ ارسال میکند

دیکدر ۳ به ۸:

کد عملیات را به سیگنال کنترلی (D0 تا D7) تبدیل می کند

مدار کنترل:

سیگنال های کنترلی را بر اساس ورودی های دریافتی تولید می کند تا عملیات مورد نظر انجام شود

توالی شمار (SC):

توالی زمانی (T0 تا T15) را تولید می کند

دیکدر ۴ به ۱۶:

سیگنال های توالی زمانی را به مدار کنترل ارسال می کنند تا عملیات در مراحل زمانی مناسب انجام شود

این شکل نشان دهنده نحوه کارکرد هماهنگ اجزای مختلف واحد کنترل برای اجرای دستورالعمل در یک کامپیوتر است

```
1 module controlsystem[
2     input  [15:0] instruction,
3     input  clk,
4     input  clr,
5     input  incr,
6     output wire [n:0] control_signals
7 ];
8     wire [15:0] IR;
9     wire [2:0] opcode = instruction[14:12];
10    wire [3:0] SC;
11    wire [7:0] D;
12    wire [15:0] T;
13
14    InstructionRegister IRModule(.instruction(instruction), .clk(clk), .IR(IR));
15    Decoder3to8 decoder3to8(.in(opcode), .out(D));
16    SequenceCounter SCModule(.clk(clk), .clr(clr), .incr(incr), .SC(SC));
17    Decoder4to16 decoder4to16(.in(SC), .out(T));
18    ControlUnit CU(.D(D), .T(T), .IR(IR), .control_signals(control_signals));
19 endmodule
```

=