

P3: Date System

Year/Month/Day/Weekday counters with BRAM

Table of Contents

1. P3_CNT_DAY.v
2. P3_CNT_MONTH.v
3. P3_CNT_YEAR.v
4. P3_CNT_WEEK.v
5. P3_leap_year.v
6. P3_weekday_calc.v
7. P3_LED_WEEK.v
8. P3_SEC1.v
9. P3_CNT_DMY.v
10. P3_TEST_CNT_DMY.v
11. P3_DMY_ref.c
12. P3_weekday.c
13. P3_weekday_coe.c

1. P3_CNT_DAY.v

File not found: /mnt/project/P3_CNT_DAY.v

2. P3_CNT_MONTH.v

```
module CNT_MONTH( RESET, CLK, CNT2, CNT10, ENABLE,
CARRY_in, CARRY_out);

input RESET, CLK, ENABLE, CARRY_in;
output reg CARRY_out;
output [3:0] CNT10;
output [3:0] CNT2;
reg [3:0] CNT10;
reg [3:0] CNT2;
reg CARRY;

always @(posedge CLK or posedge RESET)
begin
if (RESET == 1'b1) begin
CNT10 <= 4'h1;
end
else if (ENABLE == 1'b1 && CARRY_in == 1'b1) begin
if (CARRY_out == 1'b1)
CNT10 <= 4'h1;
else if (CARRY == 1'b1)
CNT10 <= 4'h0;
else
CNT10 <= CNT10 + 4'h1;
end
end

always @(CNT10 or CARRY_in or CNT2)
begin
if ((CNT10 == 4'h9 || {CNT2,CNT10} == 8'h12) && CARRY_in == 1'b1)
CARRY <= 1'b1;
else
CARRY <= 1'b0;
end

always @(CNT2 or CARRY)
begin
if (CNT2 == 4'h1 && CARRY == 1'b1)
```

```
CARRY_out <= 1'b1;  
else  
CARRY_out <= 1'b0;  
  
end  
  
always @(posedge CLK or posedge RESET)  
begin  
if (RESET == 1'b1)  
begin  
CNT2 <= 4'h0;  
end  
else if (ENABLE == 1'b1 && CARRY == 1'b1)  
begin  
if (CARRY_out == 1'b1)  
CNT2 <= 4'h0;  
else  
CNT2 <= CNT2 + 4'h1;  
end  
end  
endmodule
```

3. P3_CNT_YEAR.v

```
module CNT_YEAR( RESET, CLK, CNT10_2, CNT10, CNT2, ENABLE,
CARRY_in, CARRY_out);

input RESET, CLK, ENABLE, CARRY_in;

output reg CARRY_out;

output [3:0] CNT10;

output [3:0] CNT10_2;

output [3:0] CNT2;

reg [3:0] CNT10;

reg [3:0] CNT10_2;

reg [3:0] CNT2;

reg CARRY, CARRY_2;

always @(posedge CLK or posedge RESET)

begin

if (RESET == 1'b1) begin

CNT10 <= 4'h0;

end

else if (ENABLE == 1'b1 && CARRY_in == 1'b1) begin

if (CARRY == 1'b1)

CNT10 <= 4'h0;

else

CNT10 <= CNT10 + 4'h1;

end

end

always @(CNT10 or CARRY_in)

begin

if (CNT10 == 4'h9 && CARRY_in == 1'b1)

CARRY <= 1'b1;

else

CARRY <= 1'b0;

end

always @(CNT10_2 or CARRY)

begin

if (CNT10_2 == 4'h9 && CARRY == 1'b1)
```

```

CARRY_2 <= 1'b1;

else

CARRY_2 <= 1'b0;

end

always @(posedge CLK or posedge RESET)
begin

if (RESET == 1'b1)

begin

CNT10_2 <= 4'h0;

end

else if (ENABLE == 1'b1 && CARRY == 1'b1)

begin

if (CARRY_2 == 1'b1)

CNT10_2 <= 4'h0;

else

CNT10_2 <= CNT10_2 + 4'h1;

end

end

always @(CNT10_2 or CARRY_2)
begin

if ({CNT2,CNT10_2} == 8'h10 && CARRY_2 == 1'b1)

CARRY_out <= 1'b1;

else

CARRY_out <= 1'b0;

end

always @(posedge CLK or posedge RESET)
begin

if (RESET == 1'b1)

begin

CNT2 <= 4'h0;

end

else if (ENABLE == 1'b1 && CARRY_2 == 1'b1)

begin

if (CARRY_out == 1'b1)

```

```
CNT2 <= 4'h0;  
else  
CNT2 <= CNT2 + 4'h1;  
end  
  
end
```

endmodule

4. P3_CNT_WEEK.v

```
module CNT_WEEK(clk, reset, day_in, week_day, ENABLE, weekday_in);  
  
input clk, reset, day_in, ENABLE;  
output reg [3:0] week_day;  
input [3:0] weekday_in;  
  
always @(posedge clk or posedge reset) begin  
if (reset == 1'b1)  
week_day <= weekday_in;  
else if (day_in == 1'b1 && ENABLE == 1'b1)  
if (week_day == 4'h6)  
week_day <= 4'h0;  
else  
week_day <= week_day + 4'h1;  
end  
  
endmodule
```

5. P3_leap_year.v

```
module leap_year(year_bcd, is_leap);

input [11:0] year_bcd;
output is_leap;

wire [3:0] ones;
wire [3:0] tens;
wire [3:0] hunds;

wire ones_div4, tens_even, div4, ones_div4_2, tens_odd;

assign ones = year_bcd[3:0];
assign tens = year_bcd[7:4];
assign hunds = year_bcd[11:8];

assign ones_div4 = (ones == 4'h0) || (ones == 4'h4) || (ones == 4'h8);
assign tens_even = ~tens[0];
assign ones_div4_2 = (ones == 4'h2) || (ones == 4'h6);
assign tens_odd = tens[0];
assign div4 = ((ones_div4 && tens_even) || (ones_div4_2 && tens_odd));

assign is_leap = div4 && ~(hunds == 4'h1);

endmodule
```

6. P3_weekday_calc.v

```
endmodule
```

7. P3_LED_WEEK.v

```
module LED_WEEK(week_day, LED);  
  input [3:0] week_day;  
  output reg [15:0] LED;  
  
  always @(week_day) begin  
    case (week_day)  
      4'b0000: LED <= 16'b0011011_0_0011100_0;  
      4'b0001: LED <= 16'b1110110_0_0011101_0;  
      4'b0010: LED <= 16'b0001111_0_0011100_0;  
      4'b0011: LED <= 16'b0101010_0_1001111_0;  
      4'b0100: LED <= 16'b0001111_0_0010111_0;  
      4'b0101: LED <= 16'b1000111_0_0000101_0;  
      4'b0110: LED <= 16'b0011011_0_1110111_0;  
    default: LED <= 16'b0011011_0_0011100_0;  
    endcase  
  end  
  
endmodule
```

8. P3_SEC1.v

```
module SEC1(CLK, RESET, ENABLE, ENABLE_kHz);

input CLK, RESET;

output ENABLE, ENABLE_kHz;

reg [26:0] tmp_count;

parameter SEC1_MAX = 125000000; // 125MHz

always @(posedge CLK)
begin

if (RESET == 1'b1)
tmp_count <= 27'h000000;

else if (ENABLE == 1'b1)
tmp_count <= 27'h000000;

else
tmp_count <= tmp_count + 27'h1;

end

assign ENABLE = (tmp_count == (SEC1_MAX - 1))? 1'b1 : 1'b0;
assign ENABLE_kHz = (tmp_count[11:0] == 12'hFFF)? 1'b1 : 1'b0;
endmodule
```

9. P3_CNT_DMY .v

File not found: /mnt/project/P3_CNT_DMY .v

10. P3_TEST_CNT_DMY.v

```
module TEST_CNT_DMY;

parameter MAX_NUM = 36890;

reg clk, reset;

reg [31:0] ref [0:MAX_NUM - 1];

reg [31:0] cnt_value, cnt_value_ref;

integer i;

reg [16:0] ok_count;

parameter CYCLE = 100;

parameter SIM_SEC1_MAX = 16;

CNT_DMY #( .SEC1_MAX(SIM_SEC1_MAX) ) i1(.RESET(reset), .CLK(clk));

always @(posedge clk)
begin
    cnt_value = { i1.week_day, i1.CNT2_2, i1.CNT10_4, i1.CNT10_3,
                  i1.CNT2, i1.CNT10_2,
                  i1.CNT4, i1.CNT10} ;
end

always #(CYCLE/2)
clk = ~clk;

initial
begin
$readmemh("ref.hex", ref);
end

initial
begin
reset = 1'b1; clk = 1'b0; ok_count = 17'b0;
cnt_value_ref = ref[0];
#(3*CYCLE) reset = 1'b0;
@(posedge i1.ENABLE);
@(negedge clk);

if (cnt_value != cnt_value_ref)begin
$display("Error at step %d: cnt_value=%X expected=%X", 0, cnt_value, ref[0]);
$display("Total OK steps = %d", ok_count);
$stop;
end
end
```

```

end

else begin
ok_count = ok_count + 1;
end

for (i = 1; i < MAX_NUM; i = i + 1)
begin
@(negedge il.ENABLE);
cnt_value_ref = ref[i];
@(posedge il.ENABLE);
@(negedge clk);
if (cnt_value != cnt_value_ref)begin
$display("Error at step %d: cnt_value=%X expected=%X", i, cnt_value, ref[i]);
$display("Total OK steps = %d", ok_count);
$stop;
end
else
ok_count = ok_count + 1;
end
$display("Total OK steps = %d", ok_count);

$finish;
end

//initial
// $monitor($time,"clk=%b reset=%b cnt_value=%b", clk, reset, cnt_value);

endmodule

```

11. P3_DMY_ref.c

```
#include <stdio.h>

int main(void) {
    int year, month, day, yy, mm, dd, max_day, week_day;
    week_day = 6;

    FILE* fp = fopen("ref.hex", "w");

    if (!fp) return 1;

    for (year = 0; year <= 100; year++) {
        for (month = 1; month < 13; month++) {
            max_day = (month == 4 || month == 6 || month == 9 || month == 11) ? 30 :
            (month == 2) ? ((year % 4 == 0) ? ((year % 100 == 0) ? ((year == 0) ? 29 : 28) : 29) : 28) :
            31;

            for (day = 1; day < max_day + 1; day++) {
                yy = ((year / 100) << 8) | (((year % 100) / 10) << 4) | (year % 10);
                mm = ((month / 10) << 4) | (month % 10);
                dd = ((day / 10) << 4) | (day % 10);

                if (week_day == 7) {
                    week_day = 0;
                }

                fprintf(fp, "%01X%03X%02X%02X\n", week_day, yy, mm, dd);
                week_day = week_day + 1;
            }
        }
    }

    fclose(fp);

    return 0;
}
```

12. P3_weekday.c

```
#include <stdio.h>

int zeller(int, int, int);
int isleap(int);
int daysofmonth(int, int);

int main(void) {
    int year, month, w;

    FILE* fp = fopen("weekday.txt", "w");

    for (year = 2000; year <= 2100; year++) {
        for (month = 1; month <= 12; month++) {
            w = zeller(year, month, 1);
            fprintf(fp, "%X\n", w);
        }
    }

    fclose(fp);

    return 0;
}

int zeller(int y, int m, int d) {
    int i, j, goukei, w;
    goukei = 0;

    for (i = 1; i < y; i++) {
        for (j = 1; j <= 12; j++) {
            goukei += daysofmonth(i, j);
        }
    }

    for (j = 1; j < m; j++) {
        goukei += daysofmonth(y, j);
    }

    w = (goukei + d) % 7;

    return w;
}

int daysofmonth(int year, int month)
{
```

```
int day;

switch(month){

case 1:

case 3:

case 5:

case 7:

case 8:

case 10:

case 12: day = 31;

break;

case 4:

case 6:

case 9:

case 11: day = 30;

break;

case 2: day = 28 + isleap(year);

break;

}

return day;

}

int isleap(int year)

{

return (year % 4 == 0 && year % 100 != 0 || year % 400 == 0);

}
```

13. P3_weekday_coe.c

```
#include <stdio.h>

int main(void)
{
    int val, count;
    count = 0;
    FILE *fin = fopen("weekday.txt", "r");
    FILE *fout = fopen("weekday.coe", "w");

    fprintf(fout, "memory_initialization_radix=16;\n");
    fprintf(fout, "memory_initialization_vector=\n");

    while (fscanf(fin, "%d", &val) == 1) {
        fprintf(fout, "%d", val);
        count++;
        if (count < 1212) {
            fprintf(fout, ",");
        }
        if (count % 16 == 0) fprintf(fout, "\n");
    }

    fprintf(fout, ";\n");

    fclose(fin);
    fclose(fout);

    return 0;
}
```