

main fpu / fdiv / fdiv2.v Go to file ...

Laplace-LT Add files via upload Latest commit f8db30e 20 days ago History

1 contributor

266 lines (235 sloc) 9.14 KB Raw Blame

```
1  `timescale 1us / 100ns
2  `default_nettype none
3  module fdiv2
4      (input wire [31:0] op1,
5       input wire [31:0] op2,
6       output reg [31:0] result,
7       input wire clk,
8       input wire reset
9      );
10
11     wire [24:0] fra1;
12     wire [24:0] fra2;
13     wire [8:0] exp1;
14     wire [8:0] exp2;
15     wire sig1;
16     wire sig2;
17
18
19     assign fra1 = (op1[30:23] == 8'b0) ? {2'b00, op1[22:0]} : {2'b01, op1[22:0]};
20     assign fra2 = (op2[30:23] == 8'b0) ? {2'b00, op2[22:0]} : {2'b01, op2[22:0]};
21     assign exp1 = {1'b0, op1[30:23]};
22     assign exp2 = {1'b0, op2[30:23]};
23     assign sig1 = op1[31];
24     assign sig2 = op2[31];
25
26     reg [24:0] fra2_2;
27     reg [24:0] fra2_3;
28     reg [24:0] fra2_4;
29     reg [24:0] fra2_5;
30     reg [24:0] fra2_6;
31
32     reg ans_sig_reg_2;
33     reg ans_sig_reg_3;
34     reg ans_sig_reg_4;
35     reg ans_sig_reg_5;
36     reg ans_sig_reg_6;
37
38     wire [3:0] ans1_6;
39     wire [3:0] ans2_6;
40     wire [3:0] ans3_6;
41     wire [3:0] ans4_6;
42     wire [3:0] ans5_6;
43     wire [3:0] ans6_6;
44
45     reg [3:0] ans1_6_2;
46     reg [3:0] ans1_6_3;
47     reg [3:0] ans1_6_4;
48     reg [3:0] ans1_6_5;
49     reg [3:0] ans1_6_6;
50     reg [24:0] x4_reg;
51
52     reg [3:0] ans2_6_3;
53     reg [3:0] ans2_6_4;
54     reg [3:0] ans2_6_5;
55     reg [3:0] ans2_6_6;
56     reg [24:0] x8_reg;
57
58     reg [3:0] ans3_6_4;
59     reg [3:0] ans3_6_5;
60     reg [3:0] ans3_6_6;
61     reg [24:0] x12_reg;
62
63     reg [3:0] ans4_6_5;
64     reg [3:0] ans4_6_6;
65     reg [24:0] x16_reg;
```

```

67     reg [3:0] ans5_6_6;
68     reg [24:0] x20_reg;
69
70 //x / fra2 = ans ... amari
71     wire [24:0] sub1 = fra1 - fra2;
72     wire [24:0] x1 = (sub1[24] == '1'b1) ? (fra1 << 1) : (sub1 << 1);
73     assign ans1_6[3] = (sub1[24] == '1'b1) ? '1'b0 : '1'b1;
74     wire [24:0] sub2 = x1 - fra2;
75     wire [24:0] x2 = (sub2[24] == '1'b1) ? (x1 << 1) : (sub2 << 1);
76     assign ans1_6[2] = (sub2[24] == '1'b1) ? '1'b0 : '1'b1;
77     wire [24:0] sub3 = x2 - fra2;
78     wire [24:0] x3 = (sub3[24] == '1'b1) ? (x2 << 1) : (sub3 << 1);
79     assign ans1_6[1] = (sub3[24] == '1'b1) ? '1'b0 : '1'b1;
80     wire [24:0] sub4 = x3 - fra2;
81     wire [24:0] x4 = (sub4[24] == '1'b1) ? (x3 << 1) : (sub4 << 1);
82     assign ans1_6[0] = (sub4[24] == '1'b1) ? '1'b0 : '1'b1;
83
84
85     wire [24:0] sub5 = x4_reg - fra2_2;
86     wire [24:0] x5 = (sub5[24] == '1'b1) ? (x4_reg << 1) : (sub5 << 1);
87     assign ans2_6[3] = (sub5[24] == '1'b1) ? '1'b0 : '1'b1;
88     wire [24:0] sub6 = x5 - fra2_2;
89     wire [24:0] x6 = (sub6[24] == '1'b1) ? (x5 << 1) : (sub6 << 1);
90     assign ans2_6[2] = (sub6[24] == '1'b1) ? '1'b0 : '1'b1;
91     wire [24:0] sub7 = x6 - fra2_2;
92     wire [24:0] x7 = (sub7[24] == '1'b1) ? (x6 << 1) : (sub7 << 1);
93     assign ans2_6[1] = (sub7[24] == '1'b1) ? '1'b0 : '1'b1;
94     wire [24:0] sub8 = x7 - fra2_2;
95     wire [24:0] x8 = (sub8[24] == '1'b1) ? (x7 << 1) : (sub8 << 1);
96     assign ans2_6[0] = (sub8[24] == '1'b1) ? '1'b0 : '1'b1;
97
98
99     wire [24:0] sub9 = x8_reg - fra2_3;
100    wire [24:0] x9 = (sub9[24] == '1'b1) ? (x8_reg << 1) : (sub9 << 1);
101    assign ans3_6[3] = (sub9[24] == '1'b1) ? '1'b0 : '1'b1;
102    wire [24:0] sub10 = x9 - fra2_3;
103    wire [24:0] x10 = (sub10[24] == '1'b1) ? (x9 << 1) : (sub10 << 1);
104    assign ans3_6[2] = (sub10[24] == '1'b1) ? '1'b0 : '1'b1;
105    wire [24:0] sub11 = x10 - fra2_3;
106    wire [24:0] x11 = (sub11[24] == '1'b1) ? (x10 << 1) : (sub11 << 1);
107    assign ans3_6[1] = (sub11[24] == '1'b1) ? '1'b0 : '1'b1;
108    wire [24:0] sub12 = x11 - fra2_3;
109    wire [24:0] x12 = (sub12[24] == '1'b1) ? (x11 << 1) : (sub12 << 1);
110    assign ans3_6[0] = (sub12[24] == '1'b1) ? '1'b0 : '1'b1;
111
112
113    wire [24:0] sub13 = x12_reg - fra2_4;
114    wire [24:0] x13 = (sub13[24] == '1'b1) ? (x12_reg << 1) : (sub13 << 1);
115    assign ans4_6[3] = (sub13[24] == '1'b1) ? '1'b0 : '1'b1;
116    wire [24:0] sub14 = x13 - fra2_4;
117    wire [24:0] x14 = (sub14[24] == '1'b1) ? (x13 << 1) : (sub14 << 1);
118    assign ans4_6[2] = (sub14[24] == '1'b1) ? '1'b0 : '1'b1;
119    wire [24:0] sub15 = x14 - fra2_4;
120    wire [24:0] x15 = (sub15[24] == '1'b1) ? (x14 << 1) : (sub15 << 1);
121    assign ans4_6[1] = (sub15[24] == '1'b1) ? '1'b0 : '1'b1;
122    wire [24:0] sub16 = x15 - fra2_4;
123    wire [24:0] x16 = (sub16[24] == '1'b1) ? (x15 << 1) : (sub16 << 1);
124    assign ans4_6[0] = (sub16[24] == '1'b1) ? '1'b0 : '1'b1;
125
126
127    wire [24:0] sub17 = x16_reg - fra2_5;
128    wire [24:0] x17 = (sub17[24] == '1'b1) ? (x16_reg << 1) : (sub17 << 1);
129    assign ans5_6[3] = (sub17[24] == '1'b1) ? '1'b0 : '1'b1;
130    wire [24:0] sub18 = x17 - fra2_5;
131    wire [24:0] x18 = (sub18[24] == '1'b1) ? (x17 << 1) : (sub18 << 1);
132    assign ans5_6[2] = (sub18[24] == '1'b1) ? '1'b0 : '1'b1;
133    wire [24:0] sub19 = x18 - fra2_5;
134    wire [24:0] x19 = (sub19[24] == '1'b1) ? (x18 << 1) : (sub19 << 1);
135    assign ans5_6[1] = (sub19[24] == '1'b1) ? '1'b0 : '1'b1;
136    wire [24:0] sub20 = x19 - fra2_5;
137    wire [24:0] x20 = (sub20[24] == '1'b1) ? (x19 << 1) : (sub20 << 1);
138    assign ans5_6[0] = (sub20[24] == '1'b1) ? '1'b0 : '1'b1;
139
140
141    wire [24:0] sub21 = x20_reg - fra2_6;
142    wire [24:0] x21 = (sub21[24] == '1'b1) ? (x20_reg << 1) : (sub21 << 1);
143    assign ans6_6[3] = (sub21[24] == '1'b1) ? '1'b0 : '1'b1;
144    wire [24:0] sub22 = x21 - fra2_6;
145    wire [24:0] x22 = (sub22[24] == '1'b1) ? (x21 << 1) : (sub22 << 1);
146    assign ans6_6[2] = (sub22[24] == '1'b1) ? '1'b0 : '1'b1;
147    wire [24:0] sub23 = x22 - fra2_6;
148    wire [24:0] x23 = (sub23[24] == '1'b1) ? (x22 << 1) : (sub23 << 1);

```

```

149 assign ans6_6[1] = (sub23[24] == 1'b1) ? 1'b0 : 1'b1;
150 wire [24:0] sub24 = x23 - fra2_6;
151 // wire [24:0] x24 = (sub24[24] == 1'b1) ? (x23 << 1) : (sub24 << 1);
152 assign ans6_6[0] = (sub24[24] == 1'b1) ? 1'b0 : 1'b1;
153
154 // wire [24:0] sub_p = x_p-1 - fra2;
155 // wire [24:0] x_p = (sub_p[24] == 1'b1) ? (x_p-1 << 1) : (sub_p << 1);
156 // assign ans_high[12-p] = (sub3[24] == 1'b1) ? 1'b0 : 1'b1;
157
158 reg [8:0] ans_exp_2;
159 reg [8:0] ans_exp_3;
160 reg [8:0] ans_exp_4;
161 reg [8:0] ans_exp_5;
162 reg [8:0] ans_exp_6;
163
164 wire [8:0] for_ans_exp;
165 assign for_ans_exp = exp1 + 9'd127;
166 wire [8:0] ans_exp;
167 assign ans_exp = for_ans_exp - exp2;
168
169 wire [8:0] ans_exp_6_minus1;
170 assign ans_exp_6_minus1 = ans_exp_6 - 9'd1;
171
172
173 always @(posedge clk) begin
174     if (~reset) begin
175         result <= 32'd0;
176         ans1_6_2 <= 4'd0;
177         ans1_6_3 <= 4'd0;
178         ans1_6_4 <= 4'd0;
179         ans1_6_5 <= 4'd0;
180         ans1_6_6 <= 4'd0;
181         x4_reg <= 25'd0;
182         ans2_6_3 <= 4'd0;
183         ans2_6_4 <= 4'd0;
184         ans2_6_5 <= 4'd0;
185         ans2_6_6 <= 4'd0;
186         x8_reg <= 25'd0;
187         ans3_6_4 <= 4'd0;
188         ans3_6_5 <= 4'd0;
189         ans3_6_6 <= 4'd0;
190         x12_reg <= 25'd0;
191         ans4_6_5 <= 4'd0;
192         ans4_6_6 <= 4'd0;
193         x16_reg <= 25'd0;
194         ans5_6_6 <= 4'd0;
195         x20_reg <= 25'd0;
196
197         fra2_2 <= 25'd0;
198         fra2_3 <= 25'd0;
199         fra2_4 <= 25'd0;
200         fra2_5 <= 25'd0;
201         fra2_6 <= 25'd0;
202         ans_exp_2 <= 9'd0;
203         ans_exp_3 <= 9'd0;
204         ans_exp_4 <= 9'd0;
205         ans_exp_5 <= 9'd0;
206         ans_exp_6 <= 9'd0;
207
208         ans_sig_reg_2 <= 1'b0;
209         ans_sig_reg_3 <= 1'b0;
210         ans_sig_reg_4 <= 1'b0;
211         ans_sig_reg_5 <= 1'b0;
212         ans_sig_reg_6 <= 1'b0;
213     end else begin
214         ans1_6_2 <= ans1_6;
215         ans1_6_3 <= ans1_6_2;
216         ans1_6_4 <= ans1_6_3;
217         ans1_6_5 <= ans1_6_4;
218         ans1_6_6 <= ans1_6_5;
219         x4_reg <= x4;
220         ans2_6_3 <= ans2_6;
221         ans2_6_4 <= ans2_6_3;
222         ans2_6_5 <= ans2_6_4;
223         ans2_6_6 <= ans2_6_5;
224         x8_reg <= x8;
225         ans3_6_4 <= ans3_6;
226         ans3_6_5 <= ans3_6_4;
227         ans3_6_6 <= ans3_6_5;
228         x12_reg <= x12;
229         ans4_6_5 <= ans4_6;
230         ans4_6_6 <= ans4_6_5;
231         x16_reg <= x16;
232         ans5_6_6 <= ans5_6;
233         x20_reg <= x20;
234         fra2_2 <= fra2_2;
235         fra2_3 <= fra2_3;
236         fra2_4 <= fra2_4;
237         fra2_5 <= fra2_5;
238         fra2_6 <= fra2_6;
239         ans_exp_2 <= ans_exp_2;
240         ans_exp_3 <= ans_exp_3;
241         ans_exp_4 <= ans_exp_4;
242         ans_exp_5 <= ans_exp_5;
243         ans_exp_6 <= ans_exp_6;
244         ans_sig_reg_2 <= ans_sig_reg_2;
245         ans_sig_reg_3 <= ans_sig_reg_3;
246         ans_sig_reg_4 <= ans_sig_reg_4;
247         ans_sig_reg_5 <= ans_sig_reg_5;
248         ans_sig_reg_6 <= ans_sig_reg_6;
249     end
250 end

```

```

232     ans5_6_6 <= ans5_6;
233     x20_reg <= x20;
234
235     fra2_2 <= fra2;
236     fra2_3 <= fra2_2;
237     fra2_4 <= fra2_3;
238     fra2_5 <= fra2_4;
239     fra2_6 <= fra2_5;
240     ans_exp_2 <= ans_exp;
241     ans_exp_3 <= ans_exp_2;
242     ans_exp_4 <= ans_exp_3;
243     ans_exp_5 <= ans_exp_4;
244     ans_exp_6 <= ans_exp_5;
245     ans_sig_reg_2 <= sig1 ^ sig2;
246     ans_sig_reg_3 <= ans_sig_reg_2;
247     ans_sig_reg_4 <= ans_sig_reg_3;
248     ans_sig_reg_5 <= ans_sig_reg_4;
249     ans_sig_reg_6 <= ans_sig_reg_5;
250     if (ans1_6_6[3] == 1'b1) begin
251         if (ans_exp_6[8] == 1'b1) begin
252             result <= 32'd0;
253         end else begin
254             result <= {ans_sig_reg_6, ans_exp_6[7:0], ans1_6_6[2:0], ans2_6_6, ans3_6_6, ans4_6_6, ans5_6_6, ans6_6};
255         end
256     end else begin
257         if (ans_exp_6_minus1[8] == 1'b1) begin
258             result <= 32'd0;
259         end else begin
260             result <= {ans_sig_reg_6, ans_exp_6_minus1[7:0], ans1_6_6[1:0], ans2_6_6, ans3_6_6, ans4_6_6, ans5_6_6, ans6_6, 1'b0}; //最終ビットは速さ
261         end
262     end
263 end
264 end
265 endmodule
266 `default_nettype wire

```