```
1    ----------------------------------------------------------------------------
     -----
2    --
3    -- Title       : ALU
4    -- Design      : ALU
5    -- Author      : Robert Bacigalupo and Tyler Ovenden
6    -- Company     :
7    --
8    ----------------------------------------------------------------------------
     -----
9    --
10   -- File        : B:\Stony
     Brook\ESE345\Project\ESE345Project\ALU\src\ALU.vhd
11   -- Generated   : Sun Oct  2 17:41:31 2022
12   -- From        : interface description file
13   -- By          : Itf2Vhdl ver. 1.22
14   --
15   ----------------------------------------------------------------------------
     -----
16   --
17   -- Description :
18   --
19   ----------------------------------------------------------------------------
     -----
20
21   --{{ Section below this comment is automatically maintained
22   --   and may be overwritten
23   --{entity {ALU} architecture {ALU_Behavior}}
24
25
26   library ieee;
27   use ieee.std_logic_1164.all;
28   use ieee.numeric_std.all;
29
30   entity ALU is
31       port(
32
33            r1 : in STD_LOGIC_VECTOR(127 downto 0);
34            r2 : in STD_LOGIC_VECTOR(127 downto 0);
35            r3 : in STD_LOGIC_VECTOR(127 downto 0);
36            instrc: in STD_LOGIC_VECTOR(24 downto 0);
37            o : out STD_LOGIC_VECTOR(127 downto 0)
38            );
39   end ALU;
40
41   --}} End of automatically maintained section
42
43   architecture ALU_Behavior of ALU is
44
45   begin
46   process(r1, r2, r3, instrc)
47
48
49
50
51   variable index : integer;   --temp variable used for storing load index
52   variable tempRes: signed (63 downto 0);    --need to be 64 for possible
     overflow
```

```
53   variable temp_int0:integer:=0;
54   variable temp_int1:integer:=0;
55   variable temp_int2:integer:=0;
56   variable temp_int3:integer:=0;
57
58
59   variable tempRes128: signed (127 downto 0);
60
61
62
63   variable testMult: std_logic_vector(130 downto 0);
64   variable testRes: std_logic_vector(130 downto 0);
65   variable testRes1: std_logic_vector(32 downto 0);
66   variable testRes2: std_logic_vector(64 downto 0);
67   variable testRes3: std_logic_vector(17 downto 0);
68
69   variable testR1: std_logic_vector (17 downto 0);
70   variable testR2: signed (16 downto 0);
71
72   variable testMult1: std_logic_vector(32 downto 0);
73   variable testMult2: std_logic_vector(64 downto 0);
74   variable rd: std_logic_vector(127 downto 0);
75
76   constant max16 : std_logic_vector(15 downto 0) := "0111111111111111";
77   constant min16 : std_logic_vector(15 downto 0) := "1000000000000000";
78   constant max32: std_logic_vector(31 downto 0):=
     "01111111111111111111111111111111";
79   constant min32: std_logic_vector(31 downto 0) := X"80000000";
80   constant max64: std_logic_vector(63 downto 0):= X"7FFFFFFFFFFFFFFF";
81   constant min64: std_logic_vector(63 downto 0) := X"8000000000000000";
82
83
84   constant max16_int: integer:= 32767;
85   constant min16_int: integer:= -32768;
86   constant max32_int : integer := 2147483647;
87   constant min32_int : integer :=  -2147483648;
88
89
90
91
92   variable counter: integer:=0;       --counter for counting ones
93   variable tempPos : integer;
94   variable temp : std_logic_vector(31 downto 0); -- general temp variable,
     currently using in ROTW
95
96
97       begin
98
99
100          --load immeditate
101          if instrc(24) = '0' then
102
103
104              o <= r1;
105
106              --convert load index into a multiple of 16
107              index := to_integer(signed(instrc(23 downto 21))) * 16;
```

```
108
109             --load immeditate
110             o(index + 15 downto index) <= instrc(20 downto 5);
111
112
113
114             --r4
115                 elsif   (instrc(24 downto 23) = "10") then
116
117
118
119                     --Signed Integer Multiply-Add Low with Saturation
120                 --y low 16-bit-fields for rs3, rs2
121                 if (instrc(22 downto 20) = "000")   then
122
123
124                     for i in 0 to 3 loop
125
126                     tempPos := 32 * i;
127
128
129                     testMult1(32 downto 0) := std_logic_vector(resize(
    signed(r3(tempPos+15 downto tempPos))* signed(r2(tempPos+15 downto tempPos
    )), 33));
130                     testRes1(32 downto 0) := std_logic_vector(resize(
    signed(testMult1(32 downto 0))+ signed(r1(tempPos+31 downto tempPos)), 33
    ));
131
132
133                     if(signed(testRes1(32 downto 0)) >   signed( max32))
    then
134
135                         o(tempPos+31 downto tempPos)<=
    "01111111111111111111111111111111";
136
137                     elsif(signed(testRes1(32 downto 0)) <  signed( min32))
    then
138                         o(tempPos+31 downto tempPos)<=
    "10000000000000000000000000000000";
139
140                     else
141
142                         o(tempPos+31 downto tempPos)<= testRes1(31 downto
    0);
143
144                      end if;
145                 end loop;
146
147
148
149                 elsif (instrc(22 downto 20) = "001")    then
150                     for i in 0 to 3 loop
151                         tempPos := 32 * i;
152
153
154
155
```

```
156                     testMult1(32 downto 0) := std_logic_vector(resize(
    signed(r3(tempPos+31 downto tempPos+16))* signed(r2(tempPos+31 downto
    tempPos+16)), 33));
157                     testRes1(32 downto 0) := std_logic_vector(resize(
    signed(testMult1(32 downto 0))+ signed(r1(tempPos+31 downto tempPos)), 33
    ));
158
159
160                         if(signed(testRes1(32 downto 0)) >   signed( max32
    )) then
161
162
163                     --  if testRes1(32) = '1' then
164                         o(tempPos+31 downto tempPos)<=
    "01111111111111111111111111111111";
165
166                     elsif(signed(testRes1(32 downto 0)) <  signed( min32))
    then
167                         o(tempPos+31 downto tempPos)<=
    "10000000000000000000000000000000";
168
169                     else
170
171                         o(tempPos+31 downto tempPos)<= testRes1(31 downto
    0);
172
173
174
175                         end if;
176
177
178                 end loop;
179
180
181
182
183                 elsif (instrc(22 downto 20) = "010")    then
184
185
186
187                     for i in 0 to 3 loop
188
189                     tempPos := 32 * i;
190
191
192                     testMult(tempPos+32 downto tempPos) :=
    std_logic_vector(resize(signed(r3(tempPos+15 downto tempPos))* signed(r2(
    tempPos+15 downto tempPos)), 33));
193                     testRes(tempPos+32 downto tempPos) := std_logic_vector
    (resize(signed(r1(tempPos+31 downto tempPos))- signed(testMult(tempPos+32
    downto tempPos)), 33));
194
195
196                         if(signed(testRes(tempPos+32 downto tempPos)) >
    signed( max32)) then
197                         o(tempPos+31 downto tempPos)<=
    "01111111111111111111111111111111";
```

```
198
199                     elsif(signed(testRes(tempPos+32 downto tempPos)) <
    signed( min32)) then
200                         o(tempPos+31 downto tempPos)<=
    "10000000000000000000000000000000";
201
202                     else
203
204                         o(tempPos+31 downto tempPos)<= testRes(tempPos+31
    downto tempPos);
205
206                      end if;
207                 end loop;
208
209
210
211             elsif (instrc(22 downto 20) = "011")    then
212
213
214
215                 for i in 0 to 3 loop
216
217                         tempPos := 32 * i;
218
219
220                     testMult(tempPos+32 downto tempPos) :=
    std_logic_vector(resize(signed(r3(tempPos+31 downto tempPos+16))* signed(
    r2(tempPos+31 downto tempPos+16)), 33));
221                     testRes(tempPos+32 downto tempPos) := std_logic_vector
    (resize(signed(r1(tempPos+31 downto tempPos))- signed(testMult(tempPos+32
    downto tempPos)), 33));
222
223
224                         if(signed(testRes(tempPos+32 downto tempPos)) >
    signed( max32)) then
225                         o(tempPos+31 downto tempPos)<=
    "01111111111111111111111111111111";
226
227                     elsif(signed(testRes(tempPos+32 downto tempPos)) <
    signed( min32)) then
228                         o(tempPos+31 downto tempPos)<=
    "10000000000000000000000000000000";
229
230                     else
231
232                         o(tempPos+31 downto tempPos)<= testRes(tempPos+31
    downto tempPos);
233
234                     end if;
235
236
237
238
239
240
241
242                 end loop;
```

```
243
244
245             elsif (instrc(22 downto 20) = "100")    then
246
247
248
249                     for i in 0 to 1 loop
250
251
252                         tempPos := 64 * i;
253
254
255                     testMult(tempPos+64 downto tempPos) :=
    std_logic_vector(resize(signed(r3(tempPos+31 downto tempPos))* signed(r2(
    tempPos+31 downto tempPos)), 65));
256                     testRes(tempPos+64 downto tempPos) := std_logic_vector
    (resize(signed(testMult(tempPos+64 downto tempPos))+ signed(r1(tempPos+63
    downto tempPos)), 65));
257
258                         if(signed(testRes(tempPos+64 downto tempPos)) >
    signed( max64)) then
259                         o(tempPos+63 downto tempPos)<=
    "01111111111111111111111111111111111111111111111111111111111111111";

260
261                     elsif(signed(testRes(tempPos+64 downto tempPos)) <
    signed( min64)) then
262                         o(tempPos+63 downto tempPos)<=
    "10000000000000000000000000000000000000000000000000000000000000000";
263
264
265                     else
266
267                         o(tempPos+63 downto tempPos)<= testRes(tempPos+63
    downto tempPos);
268
269                     end if;
270
271
272
273                 end loop;
274
275                 elsif (instrc(22 downto 20) = "101")    then
276                         for i in 0 to 1 loop
277
278
279                         tempPos := 64 * i;
280
281
282                     testMult(tempPos+64 downto tempPos) :=
    std_logic_vector(resize(signed(r3(tempPos+63 downto tempPos+32))* signed(
    r2(tempPos+63 downto tempPos+32)), 65));
283                     testRes(tempPos+64 downto tempPos) := std_logic_vector
    (resize(signed(testMult(tempPos+64 downto tempPos))+ signed(r1(tempPos+63
    downto tempPos)), 65));
284
285
```

```
286                     if(signed(testRes(tempPos+64 downto tempPos)) >
    signed( max64)) then
287                         o(tempPos+63 downto tempPos)<=
    "01111111111111111111111111111111111111111111111111111111111111111";
288
289                     elsif(signed(testRes(tempPos+64 downto tempPos)) <
    signed( min64)) then
290                         o(tempPos+63 downto tempPos)<=
    "10000000000000000000000000000000000000000000000000000000000000000";
291
292
293                     else
294
295                         o(tempPos+63 downto tempPos)<= testRes(tempPos+63
    downto tempPos);
296
297                     end if;
298
299
300
301
302                 end loop;
303
304                 elsif (instrc(22 downto 20) = "110")    then
305                         for i in 0 to 1 loop
306
307
308                         tempPos := 64 * i;
309
310
311                     testMult(tempPos+64 downto tempPos) :=
    std_logic_vector(resize(signed(r3(tempPos+31 downto tempPos))* signed(r2(
    tempPos+31 downto tempPos)), 65));
312                     testRes(tempPos+64 downto tempPos) := std_logic_vector
    (resize(signed(r1(tempPos+63 downto tempPos))- signed(testMult(tempPos+64
    downto tempPos)), 65));
313
314
315                     if(signed(testRes(tempPos+64 downto tempPos)) >
    signed( max64)) then
316                         o(tempPos+63 downto tempPos)<=
    "01111111111111111111111111111111111111111111111111111111111111111";
317
318                     elsif(signed(testRes(tempPos+64 downto tempPos)) <
    signed( min64)) then
319                         o(tempPos+63 downto tempPos)<=
    "10000000000000000000000000000000000000000000000000000000000000000";
320
321
322                     else
323
324                         o(tempPos+63 downto tempPos)<= testRes(tempPos+63
    downto tempPos);
325
326                     end if;
```

```
327
328
329
330                 end loop;
331
332
333
334
335
336
337                 elsif (instrc(22 downto 20) = "111")    then
338                             for i in 0 to 1 loop
339
340
341                         tempPos := 64 * i;
342
343
344                     testMult(tempPos+64 downto tempPos) :=
    std_logic_vector(resize(signed(r3(tempPos+63 downto tempPos+32))* signed(
    r2(tempPos+63 downto tempPos+32)), 65));
345                     testRes(tempPos+64 downto tempPos) := std_logic_vector
    (resize(signed(r1(tempPos+63 downto tempPos))- signed(testMult(tempPos+64
    downto tempPos)), 65));
346
347
348                     if(signed(testRes(tempPos+64 downto tempPos)) >
    signed( max64)) then
349                         o(tempPos+63 downto tempPos)<=
    "01111111111111111111111111111111111111111111111111111111111111111";

350
351                     elsif(signed(testRes(tempPos+64 downto tempPos)) <
    signed( min64)) then
352                         o(tempPos+63 downto tempPos)<=
    "10000000000000000000000000000000000000000000000000000000000000000";
353
354
355                     else
356
357                         o(tempPos+63 downto tempPos)<= testRes(tempPos+63
    downto tempPos);
358
359                     end if;
360
361
362
363                 end loop;
364
365             end if;
366
367
368
369
370 ------------------------------------------------------------------------
    --------------
371 ----------------------r3
    instrcs-----------------------------------------------------------------
```

```
372
373                     elsif   (instrc(24 downto 23) = "11") then
374                             if instrc(18 downto 15) = "0000" then  --nop
375
376
377                                  Null;
378
379                             elsif (instrc(18 downto 15) = "0001") then
    --leading zeros
380
381                                 for i in 0 to 3 loop
382                                     tempPos := (32 * i)+31;
383                                     counter := 0;

384
385                                     for j in 0 to 31 loop
386                                         if r1(tempPos - j) = '0' then
387                                             counter := counter + 1;
388                                         else
389                                             exit;
390                                         end if;
391                                     end loop;
392                                 o((tempPos) downto (tempPos-31)) <=
    std_logic_vector(to_unsigned(counter,32));
393
394                             end loop;
395
396
397
398
399                             elsif (instrc(18 downto 15) = "0010") then
    --add word
400                                 for i in 0 to 3 loop
401                                     tempPos := 32 * i;
402                                     o(tempPos+31 downto tempPos)<=
    std_logic_vector(unsigned(r1(tempPos+31 downto tempPos)) + unsigned(r2(
    tempPos+31 downto tempPos)));
403                                 end loop;
404
405
406                             elsif (instrc(18 downto 15) = "0011") then
    --add half word
407                                 for i in 0 to 7 loop
408                                     tempPos := 16 * i;
409                                     o(tempPos+15 downto tempPos)<=
    std_logic_vector(unsigned(r1(tempPos+15 downto tempPos)) + unsigned(r2(
    tempPos+15 downto tempPos)));
410                                 end loop;
411
412
413
414                                 elsif(instrc(18 downto 15) = "0100") then
    --add half word saturated
415
416                                             for i in 0 to 7 loop
417
418                         tempPos := 16 * i;
```

```
419
420                         temp_int0 := to_integer(signed(r1(tempPos+15
    downto tempPos)))+to_integer(signed(r2(tempPos+15 downto tempPos)));
421
422                         if temp_int0>max16_int then
423                             o(tempPos+15 downto tempPos)<=
    "0111111111111111";
424                         elsif   temp_int0<min16_int then
425                             o(tempPos+15 downto tempPos)<=
    "1000000000000000";
426                         else
427                             o(tempPos+15 downto tempPos)<=
    std_logic_vector(to_signed(temp_int0,16));
428
429
430                              end if;
431                         end loop;
432
433
434
435                             elsif (instrc(18 downto 15) = "0101") then
    --and r1 r2
436                                 o <= r1 and r2;
437
438                             elsif (instrc(18 downto 15) = "0110") then  --
    broadcast word
439                                     for i in 0 to 3 loop
440                                     tempPos := 32 * i;
441                                     o((31 + tempPos) downto (tempPos)) <=
    r1(31 downto 0);
442
443                                     end loop;
444
445
446
447                             elsif (instrc(18 downto 15) = "0111") then  --
    max signed word
448                                 for i in 0 to 3 loop
449                                     tempPos := 32 * i;
450                                     temp_int1 := to_integer(signed(r1(
    tempPos+31 downto tempPos)));
451                                     temp_int2 := to_integer(signed(r2(
    tempPos+31 downto tempPos)));
452                                     if(temp_int1 > temp_int2) then
453                                         o((31 + tempPos) downto tempPos)
    <=   r1(tempPos+31 downto tempPos);
454                                     else
455                                         o((31 + tempPos) downto tempPos)
    <=   r2(tempPos+31 downto tempPos);
456                                     end if;
457                                 end loop;
458
459
460
461                             elsif (instrc(18 downto 15) = "1000") then  --
    min signed word
462                                 for i in 0 to 3 loop
```

```
463                                     tempPos := 32 * i;
464                                     temp_int1 := to_integer(signed(r1(
    tempPos+31 downto tempPos)));
465                                     temp_int2 := to_integer(signed(r2(
    tempPos+31 downto tempPos)));
466                                     if(temp_int1 < temp_int2) then
467                                         o((31 + tempPos) downto tempPos)
    <=   r1(tempPos+31 downto tempPos);
468                                     else
469                                         o((31 + tempPos) downto tempPos)
    <=   r2(tempPos+31 downto tempPos);
470                                     end if;
471                                 end loop;
472
473
474                             elsif (instrc(18 downto 15) = "1001") then
    --multiply low bits of r1, r2
475                                     for i in 0 to 3 loop
476                                     tempPos := 32 * i;
477                                     o((31 + tempPos) downto (tempPos)) <=
    std_logic_vector(unsigned(r1((15 + tempPos) downto (tempPos))) * unsigned(
    r2((15 + tempPos) downto (tempPos))));
478                                     end loop;
479
480                             elsif (instrc(18 downto 15) = "1010") then
    --multiply by constant
481                                     for i in 0 to 3 loop
482                                     tempPos := 32 * i;
483                                     o((31 + tempPos) downto (tempPos)) <=
    std_logic_vector(unsigned(r1((15 + tempPos) downto (tempPos))) * resize(
    unsigned(instrc(14 downto 10)),16));
484                                     --switch out if need to resize
485                             --          o((31 + tempPos) downto (tempPos)) <=
    std_logic_vector(unsigned(r1((15 + tempPos) downto (tempPos))) *
    unsigned(instrc(14 downto 10)));
486                                     end loop;
487
488                             elsif (instrc(18 downto 15) = "1011") then
    --or r1 r2
489                                 o <=  r1 or r2;

490
491                             elsif (instrc(18 downto 15) = "1100") then
    --counts 1s in word
492
493                                 for i in 0 to 3 loop
494                                     tempPos := 32 * i;
495                                     counter := 0;
496                                     for j in 0 to 31 loop
497                                         if r1(j + tempPos) = '1' then
498                                             counter := counter + 1;
499                                         end if;
500                                     end loop;
501                                 o((31 + tempPos) downto (tempPos)) <=
    std_logic_vector(to_unsigned(counter,32));
502
503                             end loop;
```

```
504
505                             elsif (instrc (18 downto 15) = "1101") then
    --ROTW rotate bits in word
506                                 for i in 0 to 3 loop
507                                     tempPos := 32 * i;
508                                     temp := r1(temppos + 31 downto temppos
    );
509                                     temp_int1 := to_integer(unsigned(r2(
    temppos + 5 downto temppos)));
510                                     o(temppos+31 downto temppos)<=
    std_logic_vector(unsigned(temp) ror temp_int1);
511
512                                 end loop;
513
514                             elsif (instrc(18 downto 15) = "1110") then
    --sub word
515
516
517
518                                     for i in 0 to 3 loop
519                                         tempPos := 32 * i;
520                                         o(tempPos+31 downto tempPos)<=
    std_logic_vector(unsigned(r1(tempPos+31 downto tempPos)) - unsigned(r2(
    tempPos+31 downto tempPos)));
521                                     end loop;
522
523
524                             elsif(instrc(18 downto 15) = "1111") then
    --sub saturated
525
526
527
528
529                                                 for i in 0 to 7 loop
530
531                         tempPos := 16 * i;
532
533                         temp_int0 := to_integer(signed(r1(tempPos+15
    downto tempPos)))-to_integer(signed(r2(tempPos+15 downto tempPos)));
534
535                         if temp_int0>max16_int then
536                             o(tempPos+15 downto tempPos)<=
    "0111111111111111";
537                         elsif   temp_int0<min16_int then
538                             o(tempPos+15 downto tempPos)<=
    "1000000000000000";
539                         else
540                             o(tempPos+15 downto tempPos)<=
    std_logic_vector(to_signed(temp_int0,16));
541
542
543                             end if;
544                         end loop;
545
546                 end if;
547             end if;
548
```

```
549
550
551       end process;
552   end ALU_Behavior;
553
```