```
 1  library ieee;
 2
 3  use ieee.std_logic_1164.all;
 4  use ieee.std_logic_unsigned.all;
 5
 6  entity UnitatDeControl is
 7
 8  Port ( Reset : in STD_LOGIC;
 9     clk : in STD_LOGIC;
10     co : in STD_LOGIC_VECTOR (5 downto 0);
11     func : in STD_LOGIC_VECTOR (5 downto 0);
12     z : in STD_LOGIC;
13     c : in STD_LOGIC;
14     l_sor1 : out STD_LOGIC;
15     l_sor2 : out STD_LOGIC;
16     e_reg : out STD_LOGIC;
17     mux_dest : out STD_LOGIC_VECTOR (1 downto 0);
18     l_mem : out STD_LOGIC;
19     e_mem : out STD_LOGIC;
20     e_mar : out STD_LOGIC;
21     l_mdr : out STD_LOGIC;
22     e_mdr : out STD_LOGIC;
23     e_ir : out STD_LOGIC;
24     l_pc : out STD_LOGIC;
25     e_pc : out STD_LOGIC;
26     pc_sup : out STD_LOGIC;
27     inc_pc : out STD_LOGIC;
28     l_desp1 : out STD_LOGIC;
29     l_desp2 : out STD_LOGIC;
30     l_extsign : out STD_LOGIC;
31     l_acc : out STD_LOGIC;
32     e_acc : out STD_LOGIC;
33     Tancar : out STD_LOGIC;
34     op_alu : out STD_LOGIC_VECTOR (2 downto 0)
35     );
36
37  end UnitatDeControl;
38
39  architecture Behavioral of UnitatDeControl is
40
41  type Tipus_Estats is (E0, E1, E1a, E2, E3, E4, E5, E6, E6a, E7, E8, E9, E9a, E10,
     E11, E11i, E12, E12i, E13, E14, E14i, E15, E15i, E16, E17, E18, E19, E20);
42
43  signal Estat : Tipus_Estats;
44
45  begin
46
47     transicions: process(clk)
48        begin
49
50        if falling_edge(clk) then
51           if reset='1' then
52              Estat <= E0;
53           else
54              case Estat is
55                 when E0 => Estat <= E1;
56                 when E1 => Estat <= E1a;
57                 when E1a =>
58                    -- lw o sw
59                    if co="100011" or co="101011" then
60                       Estat <= E2;
61                    -- beq/bne
```

```
62                      elsif co="000100" or co="000101" then
63                         Estat <= E7;
64                      -- J
65                      elsif  co="000010" then
66                         Estat <= E17;
67                      -- jal
68                      elsif  co="000011" then
69                         Estat <= E18;
70                      -- jr
71                      elsif func="001000" then
72                         Estat <= E20 ;
73                      -- Arit-log
74                      else
75                         Estat <= E10;
76                   end if;
77                   when E2 => Estat <= E3;
78                   when E3 =>
79                      -- lw
80                      if co="100011" then
81                         Estat <= E4;
82                      -- sw
83                      elsif co="101011" then
84                         Estat <= E5;
85                      else
86                         Estat <= E1;
87                   end if;
88                   when E4 => Estat <= E5;
89                   when E5 => Estat <= E1;
90                   when E6 => Estat <= E6a;
91                   when E6a => Estat <= E1;
92                   when E7 => Estat <= E8;
93                   when E8 =>
94                      -- beq sense flag de 0
95                      if co="000100" and z='0' then
96                         Estat <= E1;
97                      -- bne amb flag de 0
98                      elsif co="000101" and z='1' then
99                         Estat <= E1;
100                     else
101                        Estat <= E9;
102                  end if;
103                  when E9 => Estat <= E9a;
104                  when E9a => Estat <= E1;
105                  when E10 =>
106                     -- add
107                     if func = "100000" then
108                        Estat <= E11;
109                     -- sub
110                     elsif func = "100010" then
111                        Estat <= E13;
112                     -- and
113                     elsif func = "100100" then
114                        Estat <= E14 ;
115                     -- or
116                     elsif func = "100101" then
117                        Estat <= E15;
118                     -- slt
119                     elsif  func = "101010" then
120                        Estat <= E16;
121                     -- addi
122                     elsif co = "001000" then
123                        Estat <= E11i;
```

```
124                     -- andi
125                     elsif co = "001100" then
126                        Estat <= E14i;
127                     -- ori
128                     elsif co = "001101" then
129                        Estat <= E15i;
130                  end if;
131                  -- Estats logics
132                  when E11 => Estat <= E12;
133                  when E13 => Estat <= E12;
134                  when E14 => Estat <= E12;
135                  when E15 => Estat <= E12;
136                  when E16 => Estat <= E12;
137                  -- Estats inmediats
138                  when E11i => Estat <= E12i;
139                  when E14i => Estat <= E12i;
140                  when E15i => Estat <= E12i;
141                  when E12 => Estat <= E1;
142                  when E12i => Estat <= E1;
143                  -- Estats de jump
144                  when E17 => Estat <= E1;
145                  when E18 => Estat <= E19;
146                  when E20 => Estat <= E1;
147                  when others => Estat <= E1;
148               end case;
149            end if;
150         end if;
151      end process;
152      sortides: process (Estat)
153      begin
154      case Estat is
155         when E0 =>
156            l_sor1 <= '0';
157            l_sor2 <= '0';
158            e_reg <= '0';
159            mux_dest <= "00";
160            l_mem <= '0';
161            e_mem <= '0';
162            e_mar <= '0';
163            l_mdr <= '0';
164            e_mdr <= '0';
165            e_ir <= '0';
166            l_pc <= '0';
167            e_pc <= '0';
168            pc_sup <= '0';
169            inc_pc <= '0';
170            l_desp1 <= '0';
171            l_desp2 <= '0';
172            l_extsign <= '0';
173            l_acc <= '0';
174            e_acc <= '0';
175            Tancar <= '0';
176            op_alu <= "000";
177         when E1 =>
178            l_sor1 <= '0';
179            l_sor2 <= '0';
180            e_reg <= '0';
181            mux_dest <= "00";
182            l_mem <= '1';
183            e_mem <= '0';
184            e_mar <= '1';
185            l_mdr <= '0';
```

```
186             e_mdr <= '1';
187             e_ir <= '0';
188             l_pc <= '1';
189             e_pc <= '0';
190             pc_sup <= '0';
191             inc_pc <= '0';
192             l_desp1 <= '0';
193             l_desp2 <= '0';
194             l_extsign <= '0';
195             l_acc <= '0';
196             e_acc <= '0';
197             Tancar <= '0';
198             op_alu <= "000";
199          when E1a =>
200             l_sor1 <= '0';
201             l_sor2 <= '0';
202             e_reg <= '0';
203             mux_dest <= "00";
204             l_mem <= '0';
205             e_mem <= '0';
206             e_mar <= '0';
207             l_mdr <= '1';
208             e_mdr <= '0';
209             e_ir <= '1';
210             l_pc <= '0';
211             e_pc <= '0';
212             pc_sup <= '0';
213             inc_pc <= '1';
214             l_desp1 <= '0';
215             l_desp2 <= '0';
216             l_extsign <= '0';
217             l_acc <= '0';
218             e_acc <= '0';
219             Tancar <= '0';
220             op_alu <= "000";
221          when E2 =>
222             l_sor1 <= '0';
223             l_sor2 <= '0';
224             e_reg <= '0';
225             mux_dest <= "00";
226             l_mem <= '0';
227             e_mem <= '0';
228             e_mar <= '0';
229             l_mdr <= '0';
230             e_mdr <= '0';
231             e_ir <= '0';
232             l_pc <= '0';
233             e_pc <= '0';
234             pc_sup <= '0';
235             inc_pc <= '0';
236             l_desp1 <= '0';
237             l_desp2 <= '0';
238             l_extsign <= '0';
239             l_acc <= '0';
240             e_acc <= '0';
241             Tancar <= '0';
242             op_alu <= "000";
243          when E3 =>
244             l_sor1 <= '1';
245             l_sor2 <= '0';
246             e_reg <= '0';
247             mux_dest <= "00";
```

```
248             l_mem <= '0';
249             e_mem <= '0';
250             e_mar <= '0';
251             l_mdr <= '0';
252             e_mdr <= '0';
253             e_ir <= '0';
254             l_pc <= '0';
255             e_pc <= '0';
256             pc_sup <= '0';
257             inc_pc <= '0';
258             l_desp1 <= '0';
259             l_desp2 <= '0';
260             l_extsign <= '1';
261             l_acc <= '0';
262             e_acc <= '1';
263             Tancar <= '0';
264             op_alu <= "010";
265          when E4 =>
266             l_sor1 <= '0';
267             l_sor2 <= '0';
268             e_reg <= '0';
269             mux_dest <= "00";
270             l_mem <= '1';
271             e_mem <= '0';
272             e_mar <= '1';
273             l_mdr <= '0';
274             e_mdr <= '1';
275             e_ir <= '0';
276             l_pc <= '0';
277             e_pc <= '0';
278             pc_sup <= '0';
279             inc_pc <= '0';
280             l_desp1 <= '0';
281             l_desp2 <= '0';
282             l_extsign <= '0';
283             l_acc <= '1';
284             e_acc <= '0';
285             Tancar <= '1';
286             op_alu <= "000";
287          when E5 =>
288             l_sor1 <= '0';
289             l_sor2 <= '0';
290             e_reg <= '1';
291             mux_dest <= "00";
292             l_mem <= '0';
293             e_mem <= '0';
294             e_mar <= '0';
295             l_mdr <= '1';
296             e_mdr <= '0';
297             e_ir <= '0';
298             l_pc <= '0';
299             e_pc <= '0';
300             pc_sup <= '0';
301             inc_pc <= '0';
302             l_desp1 <= '0';
303             l_desp2 <= '0';
304             l_extsign <= '0';
305             l_acc <= '0';
306             e_acc <= '0';
307             Tancar <= '0';
308             op_alu <= "000";
309          when E6 =>
```

```
310             l_sor1 <= '0';
311             l_sor2 <= '1';
312             e_reg <= '0';
313             mux_dest <= "00";
314             l_mem <= '0';
315             e_mem <= '1';
316             e_mar <= '1';
317             l_mdr <= '0';
318             e_mdr <= '1';
319             e_ir <= '0';
320             l_pc <= '0';
321             e_pc <= '0';
322             pc_sup <= '0';
323             inc_pc <= '0';
324             l_desp1 <= '0';
325             l_desp2 <= '0';
326             l_extsign <= '0';
327             l_acc <= '1';
328             e_acc <= '0';
329             Tancar <= '1';
330             op_alu <= "000";
331         when E6a =>
332             l_sor1 <= '0';
333             l_sor2 <= '1';
334             e_reg <= '0';
335             mux_dest <= "00";
336             l_mem <= '0';
337             e_mem <= '1';
338             e_mar <= '1';
339             l_mdr <= '0';
340             e_mdr <= '0';
341             e_ir <= '0';
342             l_pc <= '0';
343             e_pc <= '0';
344             pc_sup <= '0';
345             inc_pc <= '0';
346             l_desp1 <= '0';
347             l_desp2 <= '0';
348             l_extsign <= '0';
349             l_acc <= '1';
350             e_acc <= '0';
351             Tancar <= '1';
352             op_alu <= "000";
353         when E7 =>
354             l_sor1 <= '0';
355             l_sor2 <= '0';
356             e_reg <= '0';
357             mux_dest <= "00";
358             l_mem <= '0';
359             e_mem <= '0';
360             e_mar <= '0';
361             l_mdr <= '0';
362             e_mdr <= '0';
363             e_ir <= '0';
364             l_pc <= '0';
365             e_pc <= '0';
366             pc_sup <= '0';
367             inc_pc <= '0';
368             l_desp1 <= '0';
369             l_desp2 <= '0';
370             l_extsign <= '0';
371             l_acc <= '0';
```

```
372             e_acc <= '0';
373             Tancar <= '0';
374             op_alu <= "000";
375          when E8 =>
376             l_sor1 <= '1';
377             l_sor2 <= '1';
378             e_reg <= '0';
379             mux_dest <= "00";
380             l_mem <= '0';
381             e_mem <= '0';
382             e_mar <= '0';
383             l_mdr <= '0';
384             e_mdr <= '0';
385             e_ir <= '0';
386             l_pc <= '0';
387             e_pc <= '0';
388             pc_sup <= '0';
389             inc_pc <= '0';
390             l_desp1 <= '0';
391             l_desp2 <= '0';
392             l_extsign <= '0';
393             l_acc <= '0';
394             e_acc <= '0';
395             Tancar <= '0';
396             op_alu <= "110";
397          when E9 =>
398             l_sor1 <= '0';
399             l_sor2 <= '0';
400             e_reg <= '0';
401             mux_dest <= "00";
402             l_mem <= '0';
403             e_mem <= '0';
404             e_mar <= '0';
405             l_mdr <= '0';
406             e_mdr <= '0';
407             e_ir <= '0';
408             l_pc <= '1';
409             e_pc <= '0';
410             pc_sup <= '0';
411             inc_pc <= '0';
412             l_desp1 <= '1';
413             l_desp2 <= '0';
414             l_extsign <= '0';
415             l_acc <= '1';
416             e_acc <= '1';
417             Tancar <= '0';
418             op_alu <= "010";
419          when E9a =>
420             l_sor1 <= '0';
421             l_sor2 <= '0';
422             e_reg <= '0';
423             mux_dest <= "00";
424             l_mem <= '0';
425             e_mem <= '0';
426             e_mar <= '0';
427             l_mdr <= '0';
428             e_mdr <= '0';
429             e_ir <= '0';
430             l_pc <= '0';
431             e_pc <= '1';
432             pc_sup <= '0';
433             inc_pc <= '0';
```

```
434             l_desp1 <= '0';
435             l_desp2 <= '0';
436             l_extsign <= '0';
437             l_acc <= '1';
438             e_acc <= '0';
439             Tancar <= '0';
440             op_alu <= "000";
441          when E10 =>
442             l_sor1 <= '0';
443             l_sor2 <= '0';
444             e_reg <= '0';
445             mux_dest <= "00";
446             l_mem <= '0';
447             e_mem <= '0';
448             e_mar <= '0';
449             l_mdr <= '0';
450             e_mdr <= '0';
451             e_ir <= '0';
452             l_pc <= '0';
453             e_pc <= '0';
454             pc_sup <= '0';
455             inc_pc <= '0';
456             l_desp1 <= '0';
457             l_desp2 <= '0';
458             l_extsign <= '0';
459             l_acc <= '0';
460             e_acc <= '0';
461             Tancar <= '0';
462             op_alu <= "000";
463          when E11 =>
464             l_sor1 <= '1';
465             l_sor2 <= '1';
466             e_reg <= '0';
467             mux_dest <= "00";
468             l_mem <= '0';
469             e_mem <= '0';
470             e_mar <= '0';
471             l_mdr <= '0';
472             e_mdr <= '0';
473             e_ir <= '0';
474             l_pc <= '0';
475             e_pc <= '0';
476             pc_sup <= '0';
477             inc_pc <= '0';
478             l_desp1 <= '0';
479             l_desp2 <= '0';
480             l_extsign <= '0';
481             l_acc <= '0';
482             e_acc <= '1';
483             Tancar <= '0';
484             op_alu <= "010";
485          when E11i =>
486             l_sor1 <= '1';
487             l_sor2 <= '0';
488             e_reg <= '0';
489             mux_dest <= "00";
490             l_mem <= '0';
491             e_mem <= '0';
492             e_mar <= '0';
493             l_mdr <= '0';
494             e_mdr <= '0';
495             e_ir <= '0';
```

```
496             l_pc <= '0';
497             e_pc <= '0';
498             pc_sup <= '0';
499             inc_pc <= '0';
500             l_desp1 <= '0';
501             l_desp2 <= '0';
502             l_extsign <= '1';
503             l_acc <= '0';
504             e_acc <= '1';
505             Tancar <= '0';
506             op_alu <= "010";
507          when E12 =>
508             l_sor1 <= '0';
509             l_sor2 <= '0';
510             e_reg <= '1';
511             mux_dest <= "01";
512             l_mem <= '0';
513             e_mem <= '0';
514             e_mar <= '0';
515             l_mdr <= '0';
516             e_mdr <= '0';
517             e_ir <= '0';
518             l_pc <= '0';
519             e_pc <= '0';
520             pc_sup <= '0';
521             inc_pc <= '0';
522             l_desp1 <= '0';
523             l_desp2 <= '0';
524             l_extsign <= '0';
525             l_acc <= '1';
526             e_acc <= '0';
527             Tancar <= '0';
528             op_alu <= "000";
529          when E12i =>
530             l_sor1 <= '0';
531             l_sor2 <= '0';
532             e_reg <= '1';
533             mux_dest <= "00";
534             l_mem <= '0';
535             e_mem <= '0';
536             e_mar <= '0';
537             l_mdr <= '0';
538             e_mdr <= '0';
539             e_ir <= '0';
540             l_pc <= '0';
541             e_pc <= '0';
542             pc_sup <= '0';
543             inc_pc <= '0';
544             l_desp1 <= '0';
545             l_desp2 <= '0';
546             l_extsign <= '0';
547             l_acc <= '1';
548             e_acc <= '0';
549             Tancar <= '0';
550             op_alu <= "000";
551          when E13 =>
552             l_sor1 <= '1';
553             l_sor2 <= '1';
554             e_reg <= '0';
555             mux_dest <= "00";
556             l_mem <= '0';
557             e_mem <= '0';
```

```
558             e_mar <= '0';
559             l_mdr <= '0';
560             e_mdr <= '0';
561             e_ir <= '0';
562             l_pc <= '0';
563             e_pc <= '0';
564             pc_sup <= '0';
565             inc_pc <= '0';
566             l_desp1 <= '0';
567             l_desp2 <= '0';
568             l_extsign <= '0';
569             l_acc <= '0';
570             e_acc <= '1';
571             Tancar <= '0';
572             op_alu <= "110";
573         when E14 =>
574             l_sor1 <= '1';
575             l_sor2 <= '1';
576             e_reg <= '0';
577             mux_dest <= "00";
578             l_mem <= '0';
579             e_mem <= '0';
580             e_mar <= '0';
581             l_mdr <= '0';
582             e_mdr <= '0';
583             e_ir <= '0';
584             l_pc <= '0';
585             e_pc <= '0';
586             pc_sup <= '0';
587             inc_pc <= '0';
588             l_desp1 <= '0';
589             l_desp2 <= '0';
590             l_extsign <= '0';
591             l_acc <= '0';
592             e_acc <= '1';
593             Tancar <= '0';
594             op_alu <= "000";
595         when E14i =>
596             l_sor1 <= '1';
597             l_sor2 <= '0';
598             e_reg <= '0';
599             mux_dest <= "00";
600             l_mem <= '0';
601             e_mem <= '0';
602             e_mar <= '0';
603             l_mdr <= '0';
604             e_mdr <= '0';
605             e_ir <= '0';
606             l_pc <= '0';
607             e_pc <= '0';
608             pc_sup <= '0';
609             inc_pc <= '0';
610             l_desp1 <= '0';
611             l_desp2 <= '0';
612             l_extsign <= '1';
613             l_acc <= '0';
614             e_acc <= '1';
615             Tancar <= '0';
616             op_alu <= "000";
617         when E15 =>
618             l_sor1 <= '1';
619             l_sor2 <= '1';
```

```
620             e_reg <= '0';
621             mux_dest <= "00";
622             l_mem <= '0';
623             e_mem <= '0';
624             e_mar <= '0';
625             l_mdr <= '0';
626             e_mdr <= '0';
627             e_ir <= '0';
628             l_pc <= '0';
629             e_pc <= '0';
630             pc_sup <= '0';
631             inc_pc <= '0';
632             l_desp1 <= '0';
633             l_desp2 <= '0';
634             l_extsign <= '0';
635             l_acc <= '0';
636             e_acc <= '1';
637             Tancar <= '0';
638             op_alu <= "001";
639         when E15i =>
640             l_sor1 <= '1';
641             l_sor2 <= '0';
642             e_reg <= '0';
643             mux_dest <= "00";
644             l_mem <= '0';
645             e_mem <= '0';
646             e_mar <= '0';
647             l_mdr <= '0';
648             e_mdr <= '0';
649             e_ir <= '0';
650             l_pc <= '0';
651             e_pc <= '0';
652             pc_sup <= '0';
653             inc_pc <= '0';
654             l_desp1 <= '0';
655             l_desp2 <= '0';
656             l_extsign <= '1';
657             l_acc <= '0';
658             e_acc <= '1';
659             Tancar <= '0';
660             op_alu <= "001";
661         when E16 =>
662             l_sor1 <= '1';
663             l_sor2 <= '1';
664             e_reg <= '0';
665             mux_dest <= "00";
666             l_mem <= '0';
667             e_mem <= '0';
668             e_mar <= '0';
669             l_mdr <= '0';
670             e_mdr <= '0';
671             e_ir <= '0';
672             l_pc <= '0';
673             e_pc <= '0';
674             pc_sup <= '0';
675             inc_pc <= '0';
676             l_desp1 <= '0';
677             l_desp2 <= '0';
678             l_extsign <= '0';
679             l_acc <= '0';
680             e_acc <= '1';
681             Tancar <= '0';
```

```
682             op_alu <= "111";
683          when E17 =>
684             l_sor1 <= '0';
685             l_sor2 <= '0';
686             e_reg <= '0';
687             mux_dest <= "00";
688             l_mem <= '0';
689             e_mem <= '0';
690             e_mar <= '0';
691             l_mdr <= '0';
692             e_mdr <= '0';
693             e_ir <= '0';
694             l_pc <= '1';
695             e_pc <= '1';
696             pc_sup <= '1';
697             inc_pc <= '0';
698             l_desp1 <= '1';
699             l_desp2 <= '0';
700             l_extsign <= '0';
701             l_acc <= '1';
702             e_acc <= '1';
703             Tancar <= '0';
704             op_alu <= "010";
705          when E18 =>
706             l_sor1 <= '0';
707             l_sor2 <= '0';
708             e_reg <= '1';
709             mux_dest <= "10";
710             l_mem <= '0';
711             e_mem <= '0';
712             e_mar <= '0';
713             l_mdr <= '0';
714             e_mdr <= '0';
715             e_ir <= '0';
716             l_pc <= '1';
717             e_pc <= '0';
718             pc_sup <= '0';
719             inc_pc <= '0';
720             l_desp1 <= '0';
721             l_desp2 <= '0';
722             l_extsign <= '0';
723             l_acc <= '0';
724             e_acc <= '0';
725             Tancar <= '1';
726             op_alu <= "000";
727          when E19 =>
728             l_sor1 <= '0';
729             l_sor2 <= '0';
730             e_reg <= '0';
731             mux_dest <= "00";
732             l_mem <= '0';
733             e_mem <= '0';
734             e_mar <= '0';
735             l_mdr <= '0';
736             e_mdr <= '0';
737             e_ir <= '0';
738             l_pc <= '1';
739             e_pc <= '1';
740             pc_sup <= '1';
741             inc_pc <= '0';
742             l_desp1 <= '0';
743             l_desp2 <= '1';
```

```
744             l_extsign <= '0';
745             l_acc <= '1';
746             e_acc <= '1';
747             Tancar <= '0';
748             op_alu <= "010";
749          when E20 =>
750             l_sor1 <= '1';
751             l_sor2 <= '0';
752             e_reg <= '0';
753             mux_dest <= "00";
754             l_mem <= '0';
755             e_mem <= '0';
756             e_mar <= '0';
757             l_mdr <= '0';
758             e_mdr <= '0';
759             e_ir <= '0';
760             l_pc <= '0';
761             e_pc <= '1';
762             pc_sup <= '0';
763             inc_pc <= '0';
764             l_desp1 <= '0';
765             l_desp2 <= '0';
766             l_extsign <= '0';
767             l_acc <= '0';
768             e_acc <= '0';
769             Tancar <= '1';
770             op_alu <= "000";
771          end case;
772       end process;
773    end Behavioral;
```