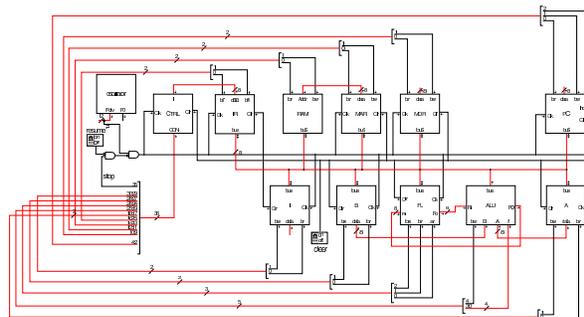


Istruzione «rotcl» e «roter» 857
 Istruzione «add_carry» 858
 Istruzione «sub_borrow» 860

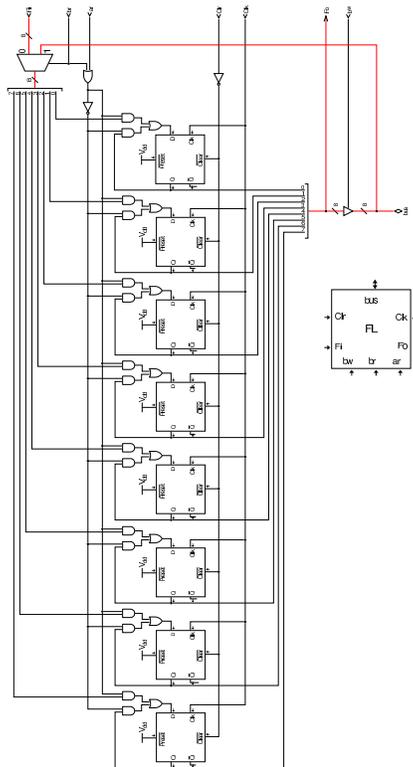
Nella quinta versione della CPU dimostrativa, viene aggiunto un registro per annotare lo stato degli indicatori, relativi all'esito di alcune operazioni svolte dalla ALU: riporto, segno, zero e straripamento.

Figura u10.1. Il bus della CPU con l'aggiunta del registro *FL* per la gestione degli indicatori.



Come si può comprendere dagli ingressi e dalle uscite che possiede, il registro *FL* può immettere dati nel bus e può essere modificato leggendo dati dal bus; inoltre, può leggere direttamente dalla ALU (ingresso *Fi*), e per questo esiste un ingresso di abilitazione ulteriore, denominato *ar* (*ALU read*), mentre fornisce in ogni istante il proprio valore memorizzato alla ALU stessa (uscita *Fo*).

Figura u10.2. La struttura interna del registro *FL*: gli otto moduli che si vedono sono flip-flop D.



Nel codice che descrive i campi del bus di controllo, si aggiungono quelli seguenti (a parte *fl_ar* già apparso nella sezione precedente), i quali servono specificatamente a gestire il registro *FL*:

```

field fl_ar[24];      // FL <-- ALU
field fl_br[25];     // FL <-- bus
field fl_bw[26];     // FL --> bus

```

Nell'elenco dei codici operativi si aggiungono istruzioni nuove e lo stesso poi nella descrizione del microcodice:

```

op move_mdr_fl {
  map move_mdr_fl : 9;          // move MDR to FL
  +0[7:0]=9;
  operands op_0;
};
op move_fl_mdr {
  map move_fl_mdr : 10;        // move FL to MDR
  +0[7:0]=10;
  operands op_0;
};
op rotcl {
  map rotcl : 42;              // A = A rotate carry left
  +0[7:0]=42;
  operands op_0;
};
op rotrc {
  map rotrc : 43;             // A = A rotate carry right
  +0[7:0]=43;
  operands op_0;
};
op add_carry {
  map add_carry : 44;         // A = A + B + carry
  +0[7:0]=44;
  operands op_0;
};
op sub_borrow {
  map sub_borrow : 45;       // A = A - B - borrow
  +0[7:0]=45;
  operands op_0;
};

```

```

begin microcode @ 0
...
//
move_mdr_fl:
  fl_br mdr_bw;           // FL <-- MDR
  ctrl_start ctrl_load;  // CNT <-- 0
//
move_fl_mdr:
  mdr_br fl_bw;          // MDR <-- FL
  ctrl_start ctrl_load;  // CNT <-- 0
//
rotcl:
  a_br alu_f=rotate_carry_left alu_bw fl_ar; // A <-- A rot. carry l
  ctrl_start ctrl_load;           // CNT <-- 0
//
rotrc:
  a_br alu_f=rotate_carry_right alu_bw fl_ar; // A <-- A rot. carry r
  ctrl_start ctrl_load;           // CNT <-- 0
//
add_carry:
  a_br alu_f=a_plus_b_carry alu_bw fl_ar; // A <-- A + B + carry
  ctrl_start ctrl_load;           // CNT <-- 0
//
sub_borrow:
  a_br alu_f=a_minus_b_borrow alu_bw fl_ar; // A <-- A - B - borrow
  ctrl_start ctrl_load;           // CNT <-- 0
...
end

```

Figura u10.6. Corrispondenza con il contenuto della memoria che rappresenta il microcodice (la coppia *m1* e *m2* dell'unità di controllo).

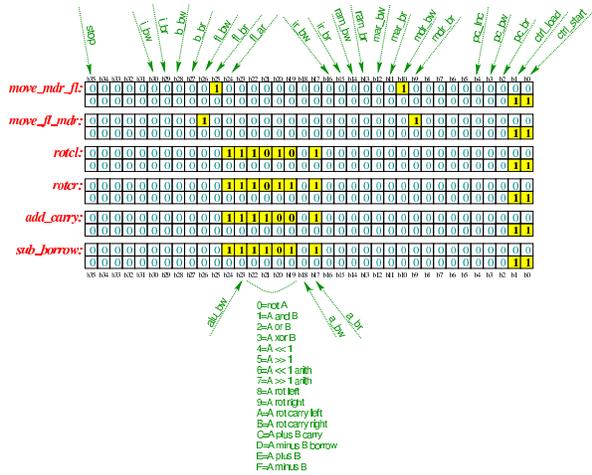


Tabella u10.7. Elenco delle macroistruzioni aggiunte in questa versione della CPU dimostrativa.

Sintassi	Descrizione
move_mdr_fl	Copia il contenuto del registro <i>MDR</i> nel registro <i>FL</i> .
move_fl_mdr	Copia il contenuto del registro <i>FL</i> nel registro <i>MDR</i> .
rotcl	Esegue la rotazione a sinistra del contenuto del registro <i>A</i> , utilizzando anche l'indicatore di riporto.
rotrc	Esegue la rotazione a destra del contenuto del registro <i>A</i> , utilizzando anche l'indicatore di riporto.
add_carry	Esegue la somma dei registri <i>A</i> e <i>B</i> , tenendo conto del riporto precedente, aggiornando di conseguenza lo stesso registro <i>A</i> .
sub_carry	Esegue la sottrazione <i>A-B</i> , tenendo conto di un'eventuale richiesta di prestito precedente, aggiornando di conseguenza lo stesso registro <i>A</i> .

Nelle sezioni successive, vengono proposti alcuni esempi, nei quali si sperimentano tutte le istruzioni nuove introdotte.

Istruzione «rotcl» e «rotrc»

Listato u10.8. Macrocodice per sperimentare le istruzioni **rotcl** e **rotrc**: si carica in memoria il valore da assegnare al registro *A*, si eseguono cinque scorrimenti a sinistra, con l'uso del riporto e il risultato viene copiato nel registro *B*; poi, con il valore presente in quel momento nel registro *A*, si eseguono altri cinque rotazioni a destra, sempre con l'uso del riporto. Il file completo che descrive le memorie per Tkgate dovrebbe essere disponibile presso allegati/circuiti-logici/scpu-sub-d-rotc.gm.

```

begin macrocode @ 0
start:
  load_imm #data_1
  move_mdr_a
  rotcl
  rotcl
  rotcl
  rotcl
  rotcl
  move_a_mdr
  move_mdr_b
  rotrc
  rotrc
  rotrc
  rotrc
  rotrc

```