

Elaborato SIS

ARCHITETTURA DEGLI ELABORATORI

CORSO DI LAUREA IN INFORMATICA, A.A. 2018/2019

Filippo Zeni (VR437690)

Giuseppe Brusco (VR429704)

Consegna

Si progetti un dispositivo per la gestione di un centro di lavoro per asportazione di truciolo (fresa). Il sistema è composto da una macchina utensile a controllo numerico (Milling Machine) e da due unità di controllo qualità poste prima dell'ingresso e dopo l'uscita del pezzo dalla macchina.

Il dispositivo di controllo prende in ingresso una sequenza di 7 bit, che assumono significato diverso in funzione dello stato in cui si trova l'impianto:

- Quando il macchinario è spento, i primi 3 bit servono per inviare il segnale di accensione, mentre i successivi non vengono considerati;
- Durante l'elaborazione, il primo bit rappresenta la qualità del pezzo ($QI=1$ buono, $QI=0$ scarto) e i successivi 5 per l'input del volume;
- Durante la fresatura, il primo bit rappresenta la riuscita o meno dell'elaborazione ($EM = 1$ buono, $EM = 0$ errore. Gli errori verranno esplicitati successivamente).
- Durante il controllo finale del pezzo lavorato, il primo bit rappresenta l'indicatore di qualità ($QO = 1$ buono, $QO=0$ scarto), mentre i successivi non vengono considerati.

Sono presenti inoltre dei contatori (NA, NB, NC, NE, NF), espressi in 4 bit, i quali vengono incrementati ogni qualvolta i corrispondenti gate (GA, GB, GC, GE, GF, ad eccezione di GD) vengono attraversati da un pezzo.

Durante la lavorazione possono essere generati degli errori, descritti di seguito:

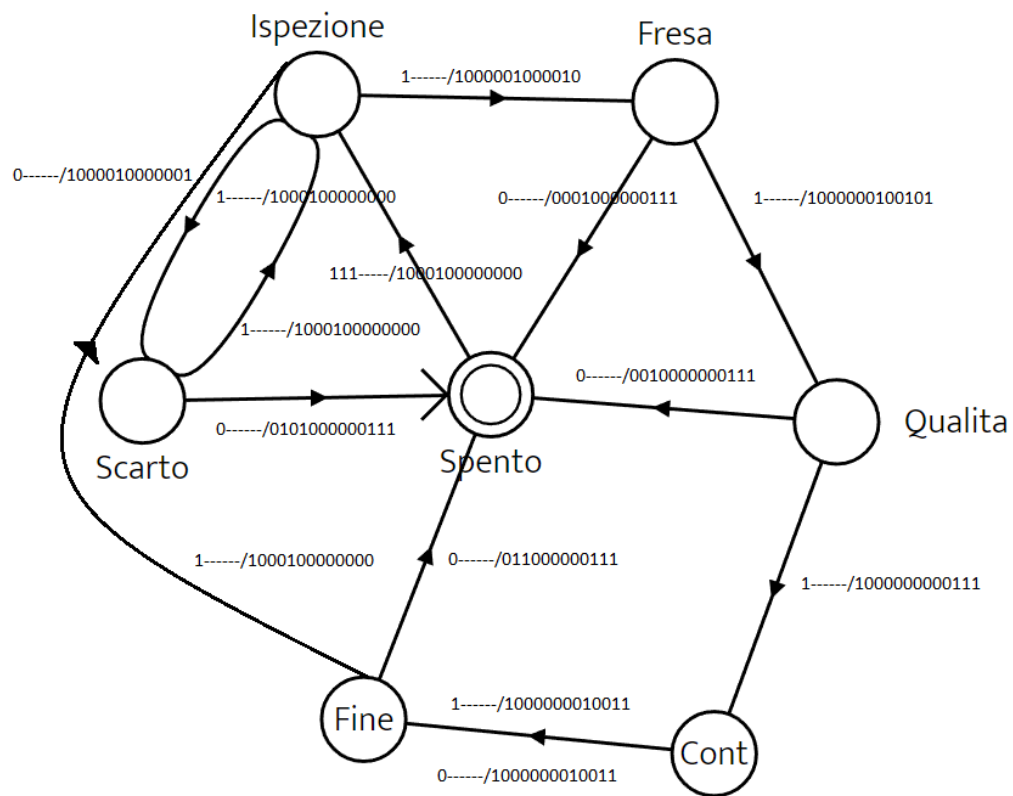
- Errore 001: l'impianto si spegne a causa della lavorazione non riuscita;
- Errore 010: l'impianto si spegne a causa del superamento della quantità di scarto di 200;
- Errore 101: l'impianto si spegne a causa di un overload del gate NB (numero di pezzi nel deposito scarti in ingresso);
- Errore 110: l'impianto si spegne a causa di un overload del gate NF (numero di pezzi nel deposito scarti in uscita);

Ad ogni ciclo di clock il dispositivo deve restituire una sequenza di 30 bit indicativi delle seguenti variabili:

O/I ERR GA GB GC GD GE GF NA NB NC NE NF

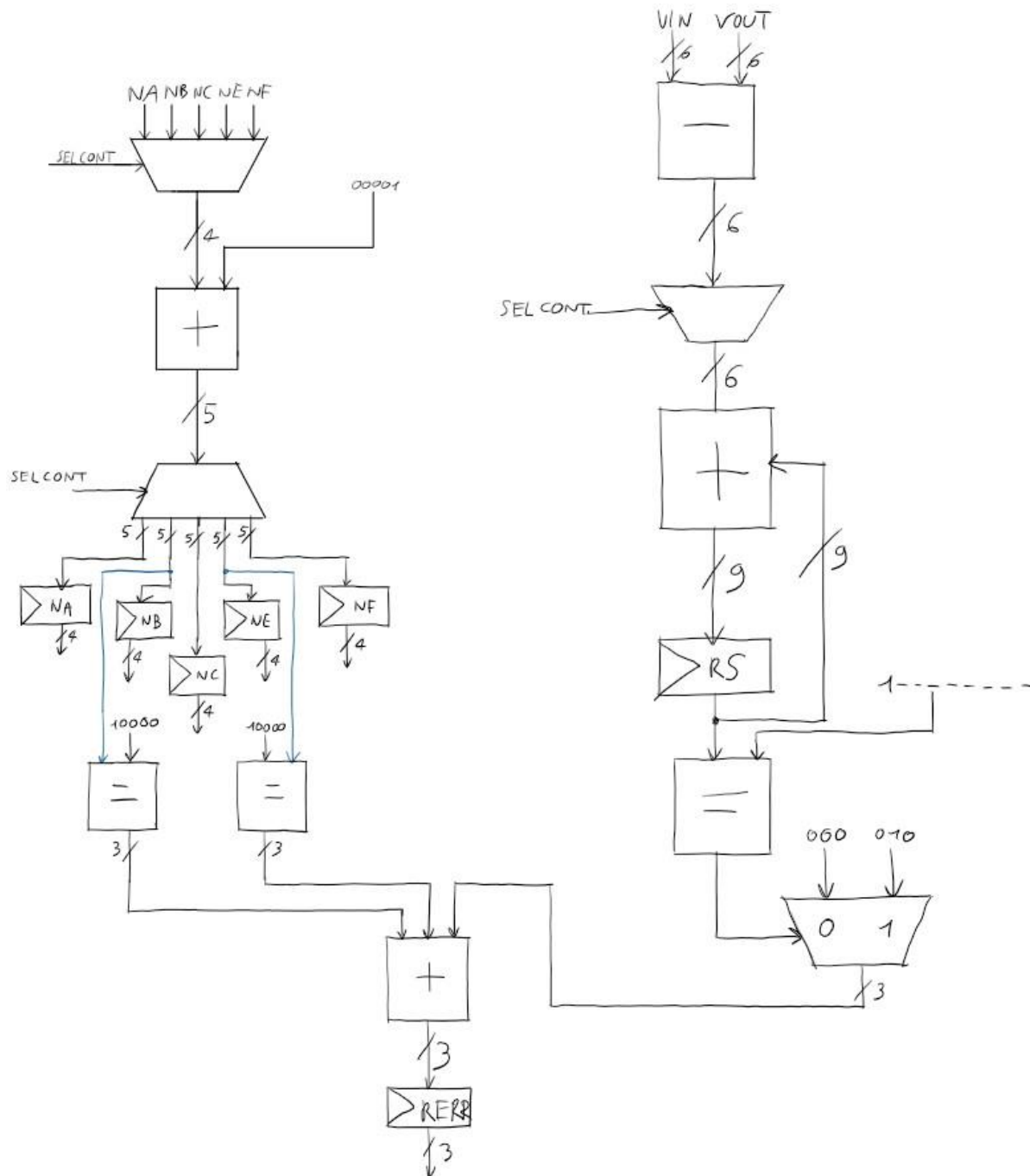
Diagramma degli stati

Schema



Datapath

Di seguito si trova la struttura del datapath realizzato:



Come si può notare, sono presenti 6 bit di input (6 per il volume e 3 per il selettore) e 23 bit di output (i primi 3 per gli errori, i successivi 20 per i contatori, uno ogni 4 bit).

Tutti i componenti presenti sono raggruppati e descritti nel seguente elenco.

Multiplexer (mux.blif)

- MUX5-4: Multiplexer a 5 ingressi da 4 bit ciascuno per selezionare il registro da incrementare;
- MUX2-5-RESET: Multiplexer a 2 ingressi da 5 bit ciascuno per impedire il reset degli altri registri a causa del DEMUX5-5;

- MUXNB: Multiplexer per gestire l'errore di overflow del contatore NB;
- MUXNE: Multiplexer per gestire l'errore di overflow del contatore NE;
- MUX2-6: Multiplexer a 2 ingressi da 6 bit ciascuno per decidere se sommare il valore calcolato;
- MUX2-9-RESET: Multiplexer a 2 ingressi da 9 bit ciascuno per il reset del volume se in overflow;
- MUXS: Multiplexer per gestire l'errore di overflow dei pezzi di scarto della macchina superiori a 200.

Sommatori (sommatori.blif)

- INCREMENTO: Sommatore che incrementa di 1 il valore precedente del contatore;
- SOMMATORE9: Sommatore a 9 bit che somma il volume calcolato e il volume contenuto nel registro;
- SOMMATORE3: Sommatore a 3 bit che somma uno dei tre errori usciti nel datapath in un giro di clock (nessuna possibilità di overflow perché uscirà sempre un solo valore alla volta). L'errore 001 non viene gestito dal datapath, ma dall'fsm per cui avendo come output dell'FSMD gli errori del datapath, l'errore 001 verrà sempre riconosciuto come 000.

Demultiplexer (demux.blif)

- DEMUX5-5: Demultiplexer a 5 bit e 5 uscite per selezione il contatore da sovrascrivere.

OR (or.blif)

- OR5= Or a 5 bit per selezionare l'uscita del multiplexer, cioè il registro in cui salvare il contatore.

Registri (registri.blif)

- REGISTRO4: Registro a 4 bit per salvare un valore a 4 bit;
- REGISTRO9: Registro a 9 bit per salvare un valore a 9 bit;
- REGISTRO6: Registro a 6 bit per salvare un valore a 6 bit;
- REGISTRO3: Registro a 3 bit per salvare un valore a 3 bit.

Uguale (uguale.blif)

- UGUALE5: Uguale a 5 bit per gestire l'overflow del contatore NB e NE;
- UGUALE9: Uguale a 9 bit per gestire l'overflow del superamento dello scarto durante la fresatura (Maggiore di 200).

Sottrattore (sottrattore.blif)

- SOTTRATTORE: Sottrattore che esegue il complemento a 1 e successivamente a 2 se il VUOT rispetto al VIN;

Statistiche del circuito

Pre ottimizzazione

Abbiamo deciso di non ottimizzare i singoli componenti, ma di lasciare il datapath costituito da componenti non ottimizzati.

Nella prima fase abbiamo ottimizzato l'FSM con i seguenti comandi:

- State_minimize stamina
- State_assign jedi: ottenendo 219 letterali e 16 nodi
- Script.rugged: ottenendo 60 letterali e 18 nodi
- Sweep: letterali e nodi uguali all'output di script.rugged

Successivamente abbiamo ottimizzato il datapath allo stesso modo:

- State_assign jedi: ottenendo 884 letterali e 211 nodi
- Script.rugged: ottenendo 376 letterali e 100 nodi
- Sweep: ottenendo 366 e 89 nodi

Post-ottimizzazione

Come post-ottimizzazione abbiamo confrontato un FSMD con datapath non ottimizzato e un FSMD con datapath ottimizzato.

FSMD con datapath non ottimizzato:

- Iniziale: 944 letterali e 229 nodi;
- Script.rugged: 423 letterali e 117 nodi;
- Sweep: 413 letterali e 106 nodi;

Mappando l'FSMD con la libreria synch.genlib e ottimizzando il circuito per area otteniamo:

- Area iniziale: 10168;
- Area finale: 9992;
- Ritardo iniziale: 34.40;
- Ritardo finale: 34.40.

FSMD con datapath ottimizzato:

- Iniziale: 426 letterali e 107 nodi;
- Script.rugged: 415 letterali e 101 nodi;
- Sweep: 415 letterali e 101 nodi;

Mappando l'FSMD con la libreria synch.genlib e ottimizzando il circuito per area otteniamo:

- Area iniziale: 10200;
- Area finale: 9832;
- Ritardo iniziale: 39.00;
- Ritardo finale: 38.60.

Descrizione delle scelte progettuali effettuate

Per verificare la correttezza dell'FSMD abbiamo creato dei file di test appositi per verificare che i tre casi di errore per overflow e il caso di errore FSM siano gestiti correttamente.