

Università degli Studi di Padova – Dipartimento di Ingegneria Industriale

Corso di Laurea in Ingegneria dell'Energia

Relazione per la prova finale

**«IMPLEMENTAZIONE VIA SOFTWARE DI UNA MACCHINA A
STATI FINITI PER DISTRIBUTORE AUTOMATICO DI BEVANDE»**

Tutor universitario: Prof. Luigi Alberti

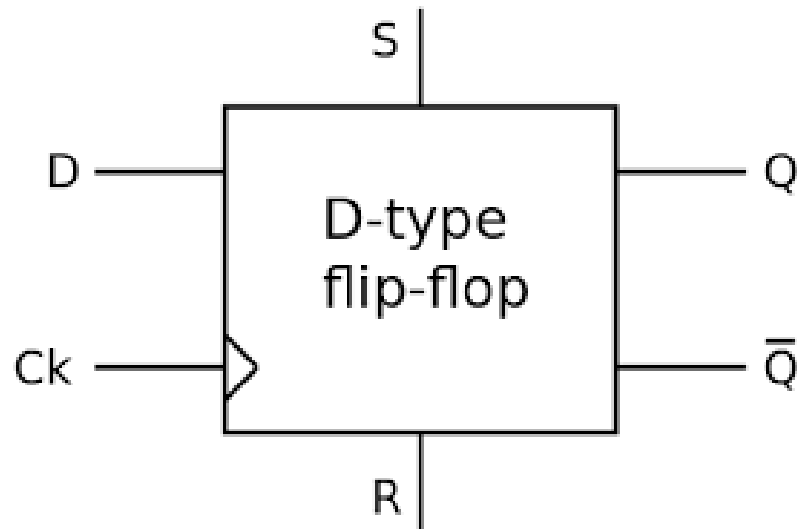
Laureando: *Matteo Multineddu*

Padova, 20/11/2023

In questo progetto vedremo due distributori automatici di bevande: uno costruito con dei flip flop di tipo D e un altro costruito attorno ad un Arduino UNO R3.

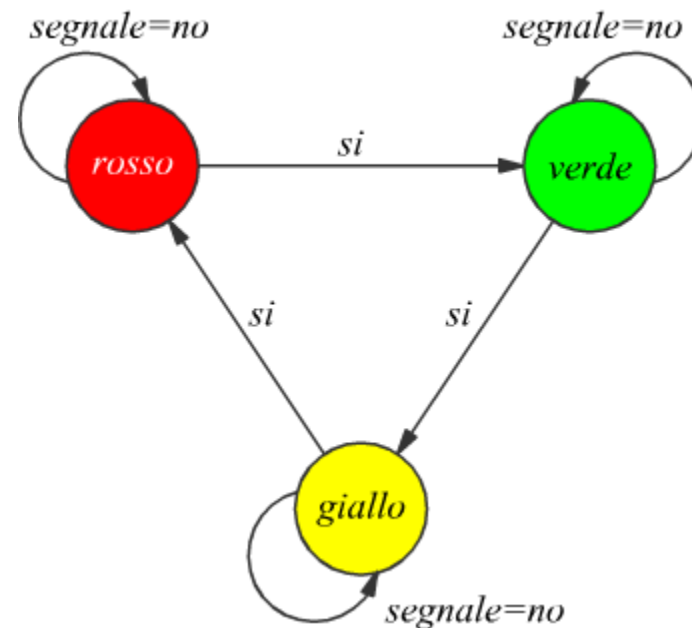
Questi due componenti sono stati scelti a causa della grande differenza di funzionalità tra loro:

- Il flip flop capace solo di memorizzare un bit
- L'Arduino capace di leggere, memorizzare, confrontare e mettere in relazione tra loro diversi tipi di dati, di conseguenza dandoci la possibilità di costruire qualcosa di molto più complesso



Una macchina a stati finiti è un modello matematico utilizzato per rappresentare un **processo** o un **comportamento sequenziale**. Queste macchine sono composte da un numero finito di stati, da qui il nome, che rappresentano «in che punto del processo ci troviamo» e a cui vengono associate azioni che l'automa poi svolgerà una volta raggiunto quello specifico stato. Il passaggio da uno stato all'altro non è casuale, è sempre necessario uno specifico input (che può variare di stato in stato) per andare avanti. I processi vengono comunemente descritti tramite **flowchart** o **grafi di stato**.

Esempio: Semaforo



Progettare e simulare due distributori con le seguenti funzionalità, già da qua possiamo vedere la quantità di restrizioni che bisognerebbe applicare per evitare di avere un circuito troppo complesso, nel caso si decidesse di usare solo dei flip flop.

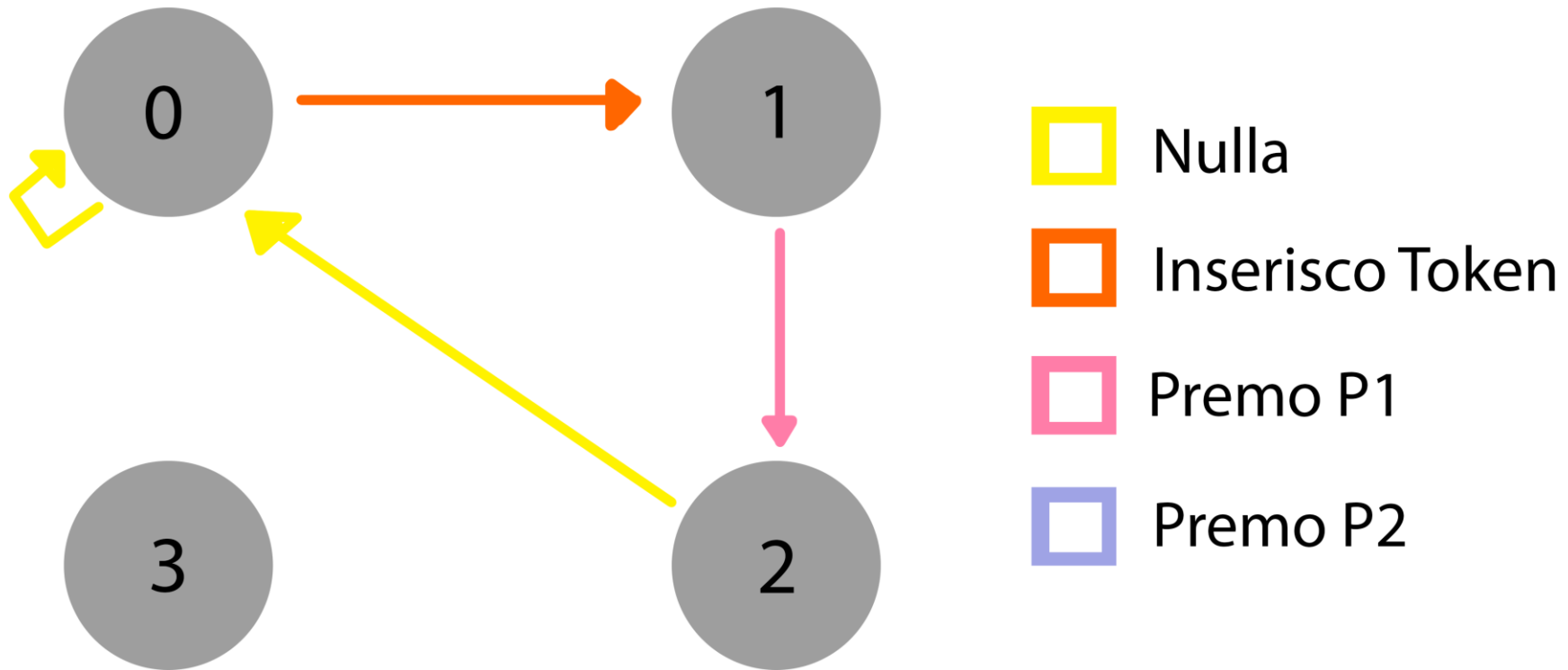
Distributore con flip flop NET tipo D

- **Accetterà una sola moneta (token)**
- **Bustine e bicchieri infiniti**
- **Zucchero non disponibile**
- Si suppone di avere un **quantitativo infinito di acqua**
- **Non fornisce resto**
- I soldi vanno inseriti PRIMA di aver scelto la bevanda
- Una volta inserito il token, **non lo si potrà più riavere indietro**
- **2 possibili bevande selezionabili**, per semplicità avranno tutte lo stesso costo (token), più precisamente saranno: espresso (P1) e thè al limone (P2)

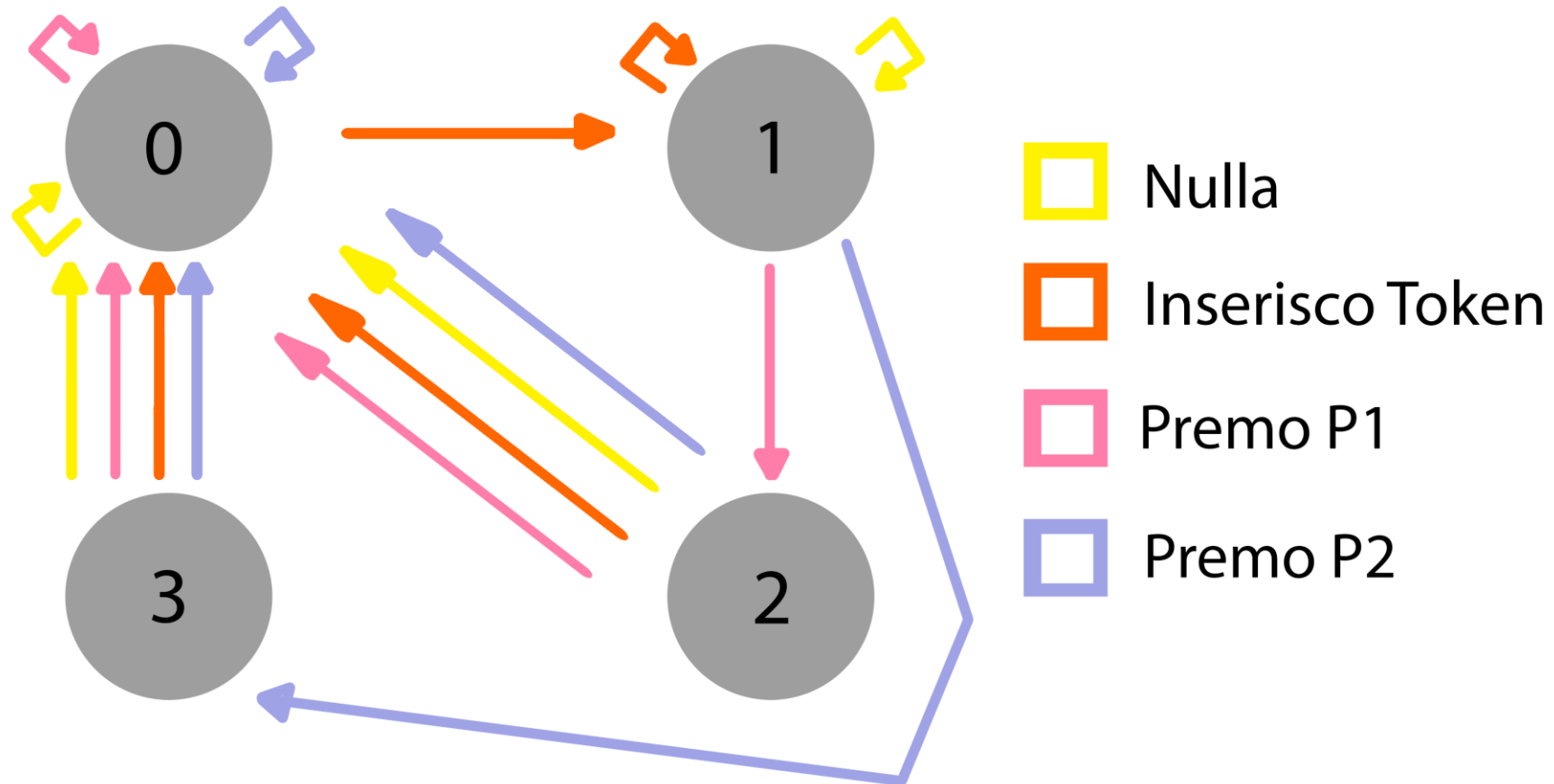
Distributore con Arduino UNO R3

- **Accettare tutti i tipi di moneta esclusi 1c, 2c e 2€ (5c, 10c, 20c, 50c, 1€) e dare resto**
- **10 bustine per ogni bevanda**, in caso di esaurimento scorte **segnerà che la bevanda non è più disponibile** e non si potrà più scegliere quella bevanda come opzione
- Si suppone di avere un quantitativo infinito di acqua e zucchero. **10 bicchieri** e 5 livelli di zucchero
- **pulsante per il reset del credito**, verranno restituiti i soldi inseriti se presenti.
- I soldi si possono inserire quando si vuole
- **5 possibili bevande**: espresso (20c), thè al limone (25c), cappuccino (30c), caffè al ginseng (35c), cappuccino al cioccolato (40c)
- Verrà specificato a quanto ammonterà il resto consegnato

IMPLEMENTAZIONE TRAMITE FLIP FLOP D NET- ESEMPIO: ORDINAZIONE CAFFÈ ESPRESSO



IMPLEMENTAZIONE TRAMITE FLIP FLOP D NET- GRAFO DI STATO COMPLESSIVO



Le tabelle di transizione sono il cuore della progettazione di macchine a stati finiti

Tabella di transizione di stato
(cosa vediamo noi)

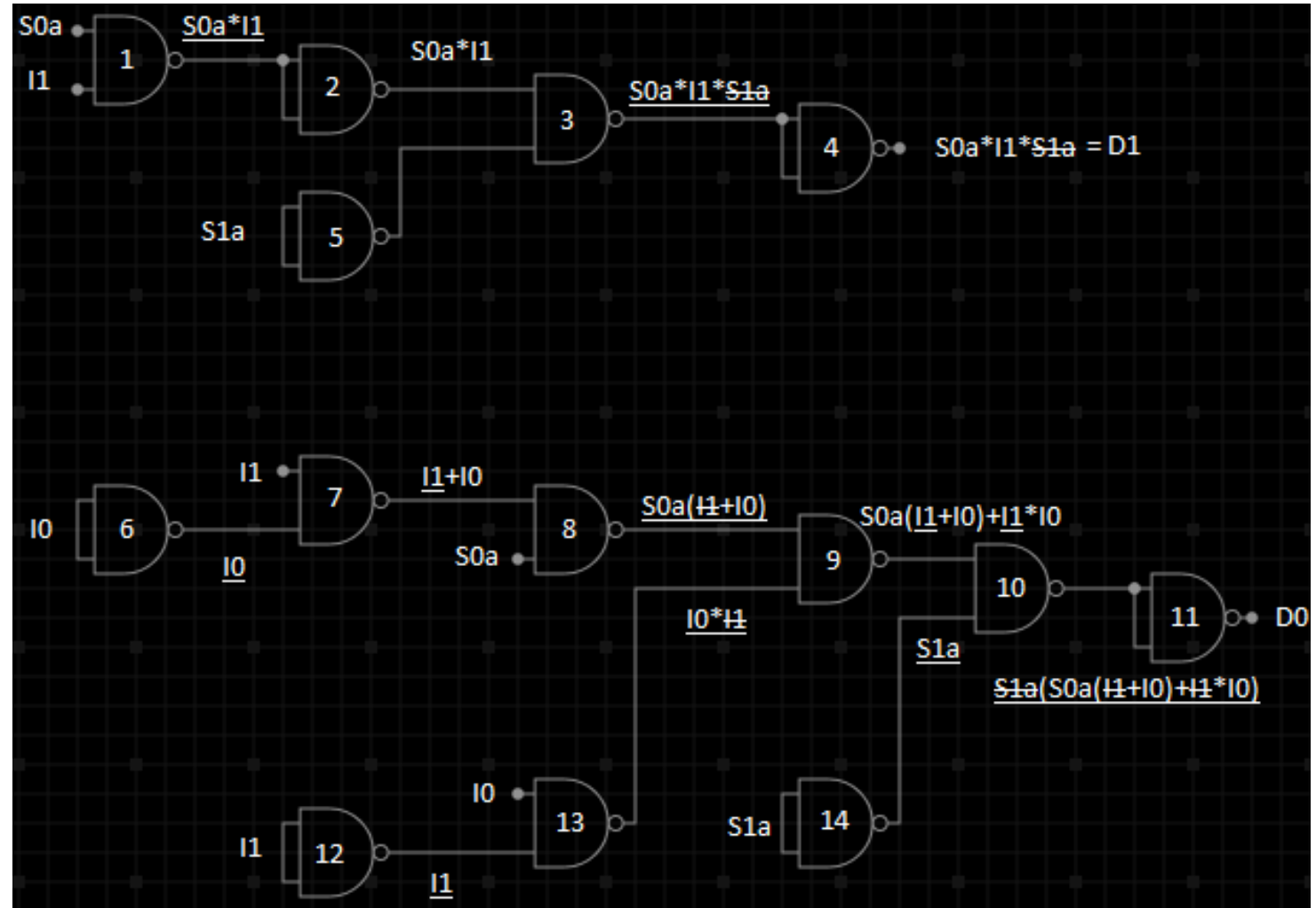
Ingresso	Stato attuale	Stato Futuro	Uscita
Nulla	S0	S0	0
Token	S0	S1	0
P1	S0	S0	0
P2	S0	S0	0
Nulla	S1	S1	0
Token	S1	S1	0
P1	S1	S2	0
P2	S1	S3	0
Nulla	S2	S0	1
Token	S2	S0	1
P1	S2	S0	1
P2	S2	S0	1
Nulla	S3	S0	1
Token	S3	S0	1
P1	S3	S0	1
P2	S3	S0	1

Tabella di transizione di stato in binario
(cosa vede il flip flop)

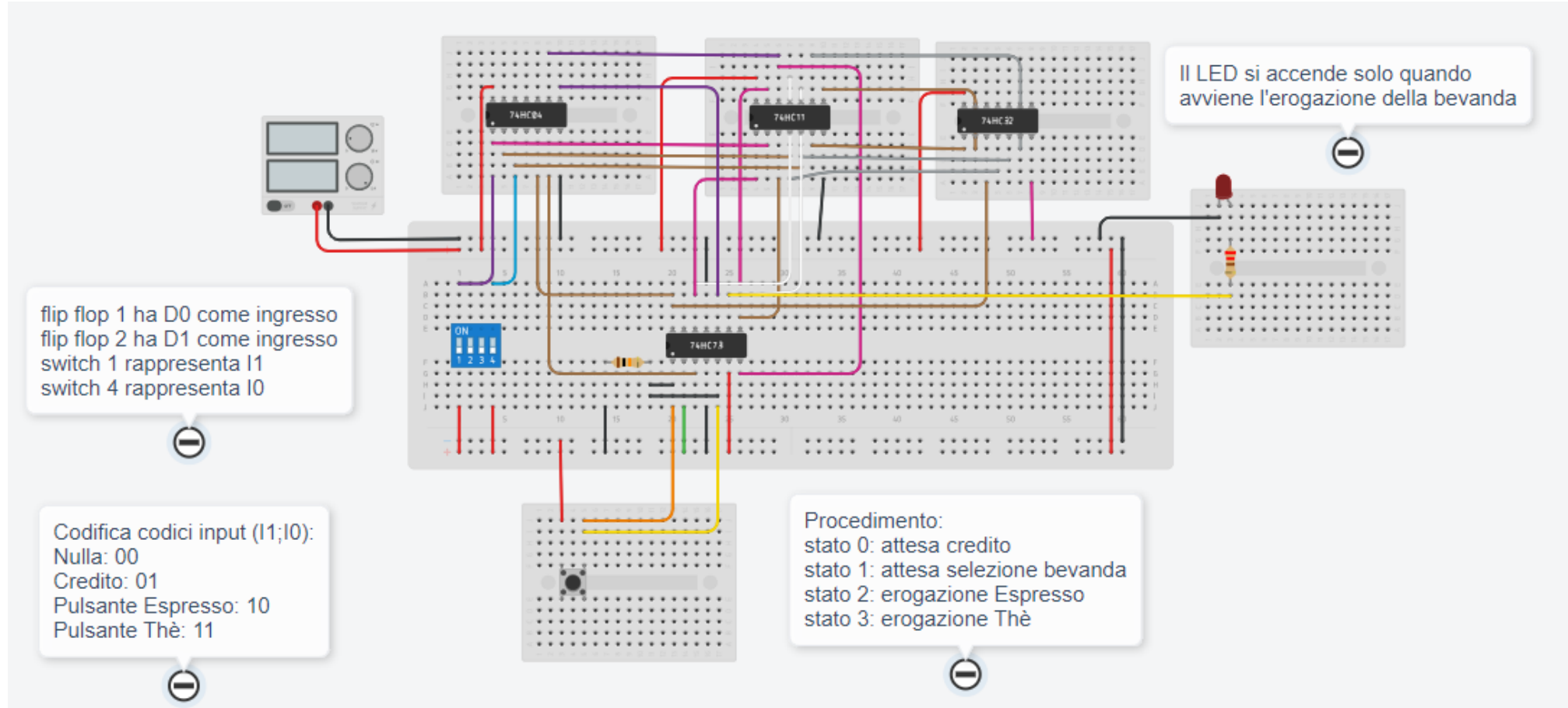
I1	I0	S1a	S0a	S1f	S0f	U
0	0	0	0	0	0	0
0	1	0	0	0	1	0
1	0	0	0	0	0	0
1	1	0	0	0	0	0
0	0	0	1	0	1	0
0	1	0	1	0	1	0
1	0	0	1	1	0	0
1	1	0	1	1	1	0
0	0	1	0	0	0	1
0	1	1	0	0	0	1
1	0	1	0	0	0	1
1	1	1	0	0	0	1
0	0	1	1	0	0	1
0	1	1	1	0	0	1
1	0	1	1	0	0	1
1	1	1	1	0	0	1

Osservazione

1. 14 NAND necessarie
2. Integrato di Tinkercad ne ha 4 al suo interno
3. Servirebbero 4 integrati tenendo due porte logiche inutilizzate
4. Usando tutte le porte logiche invece che solo le NAND verrebbero utilizzati 3 integrati, escludendo i flip flop
5. Si dimostra dunque come sia necessario più spazio nel caso di sintesi con sole porte NAND



IMPLEMENTAZIONE TRAMITE FLIP FLOP D NET- CIRCUITO FINALE SU TINKERCAD



Gli interrupt sono segnali che interrompono il normale processo di una macchina per svolgere una particolare azione da noi definita. Sono utili nei sistemi multitasking perché ci risparmiano il bisogno di sondare costantemente dato che si attivano da soli solo quando c'è necessità.

In questo progetto sono stati utilizzati 2 interrupt, entrambi di tipo hardware ed esterni. Sono però stati attivati in modo diverso: uno in linguaggio di alto livello e uno in un linguaggio di livello più basso.

```
int interruptPindue=2;  
attachInterrupt(digitalPinToInterrupt(interruptPindue)
```

```
PCICR |=B00000001;  
PCMSK0 |=B00010000;
```

PCICR – Pin Change Interrupt Control Register

Bit	7	6	5	4	3	2	1	0	
(0x68)	–	–	–	–	–	PCIE2	PCIE1	PCIE0	PCICR
Read/Write	R	R	R	R	R	R/W	R/W	R/W	
Initial Value	0	0	0	0	0	0	0	0	

Abilitazione PCMSK

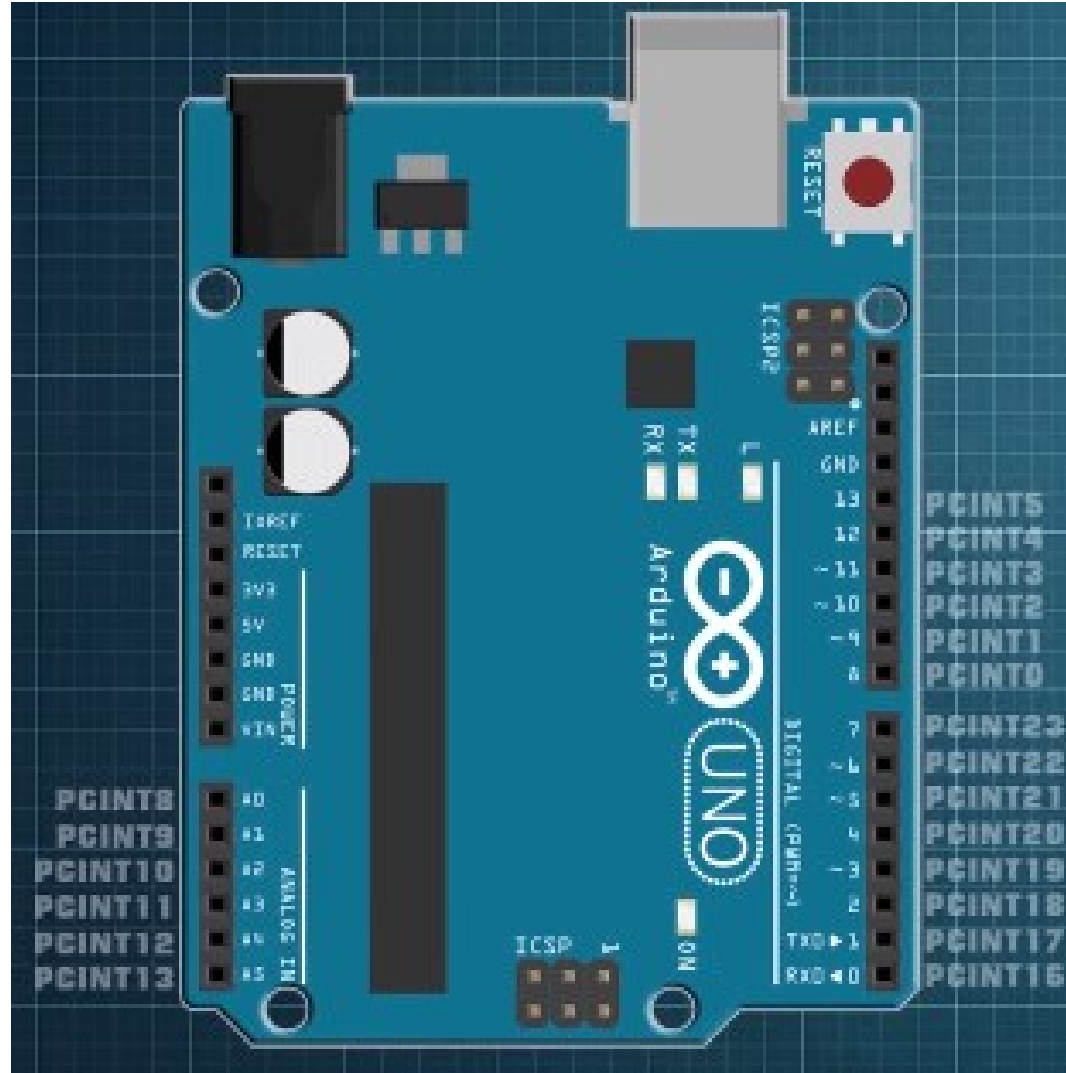
PCMSK0 – Pin Change Mask Register 0

Bit	7	6	5	4	3	2	1	0	
(0x6B)	PCINT7	PCINT6	PCINT5	PCINT4	PCINT3	PCINT2	PCINT1	PCINT0	PCMSK0
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
Initial Value	0	0	0	0	0	0	0	0	

Pin corrispondenti: Aref GND 13 12 11 10 9 8

IMPLEMENTAZIONE TRAMITE ARDUINO UNO R3- INTERRUPT

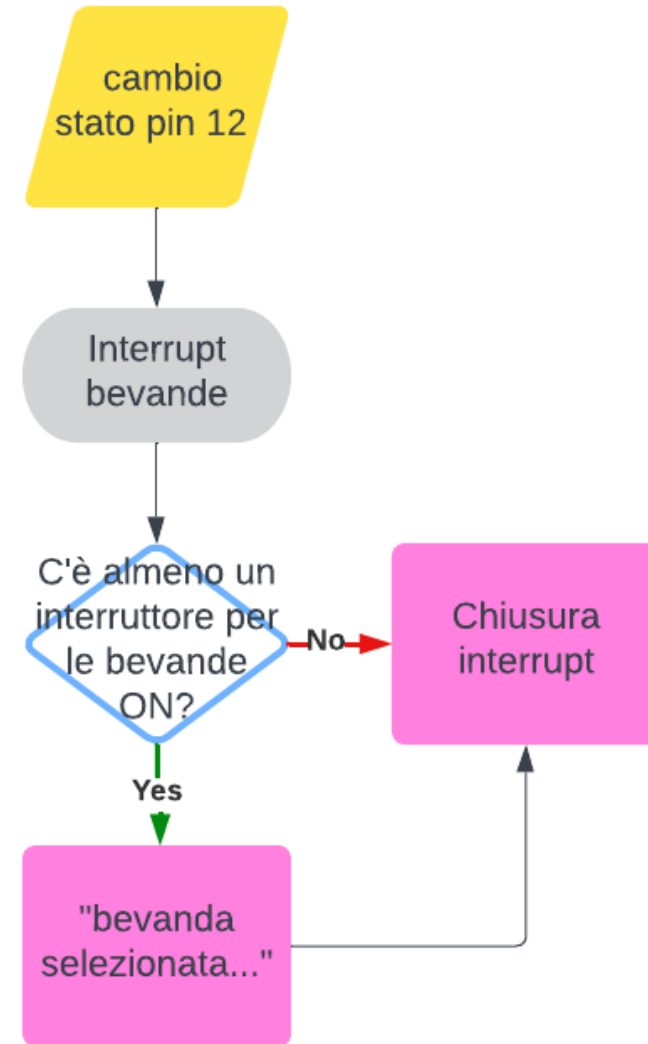
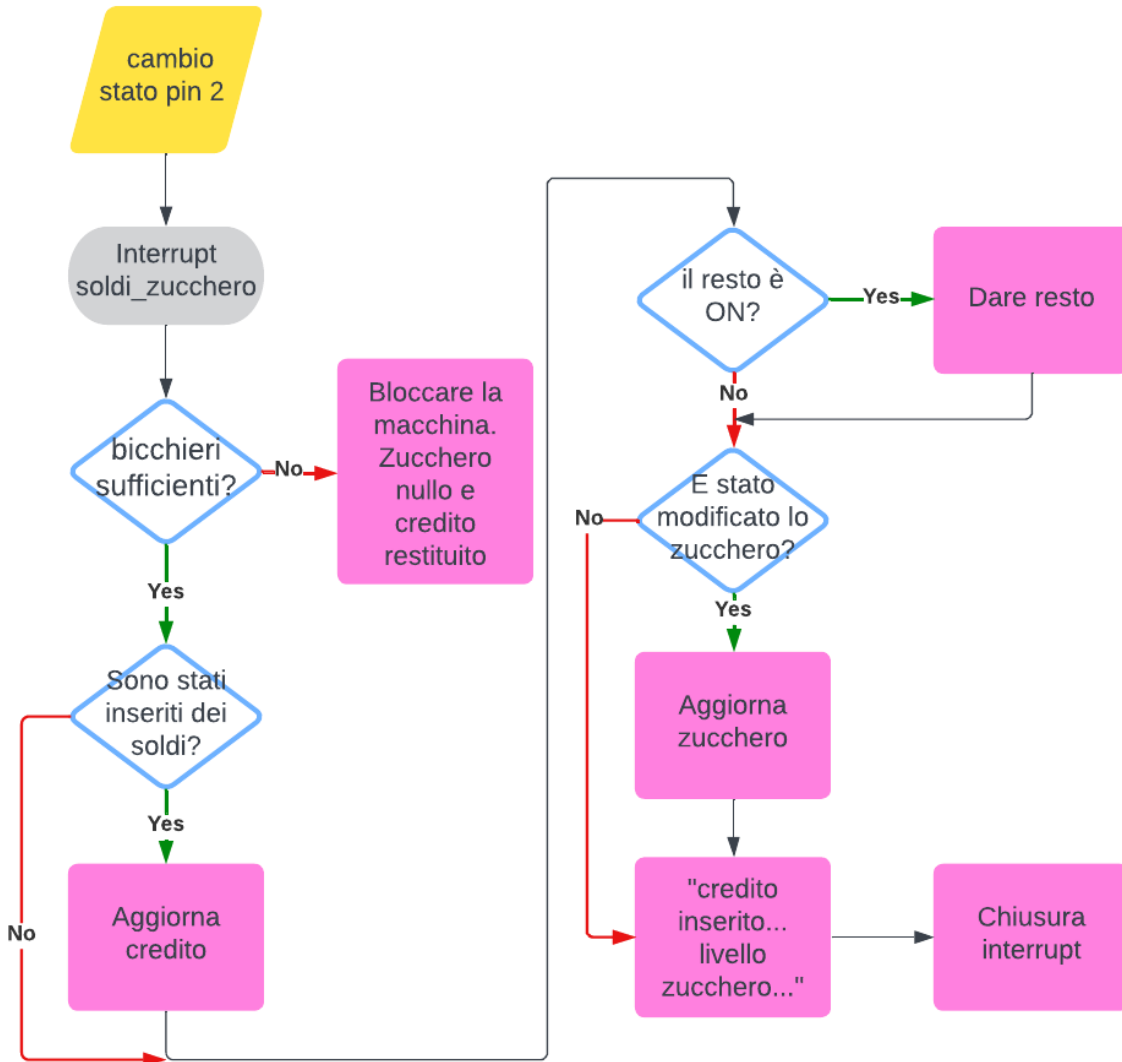
PCIE1



PCIE0

PCIE2

IMPLEMENTAZIONE TRAMITE ARDUINO UNO R3- FLOWCHART DEGLI INTERRUPT



IMPLEMENTAZIONE TRAMITE ARDUINO UNO R3-

FLOWCHART PRINCIPALE

