

# Corso di Calcolatori Elettronici

## Introduzione al Corso

Roberto Canonico

Università degli Studi di Napoli Federico II

A.A. 2022-2023





Fornire le conoscenze di base necessarie relative a:

- **Reti Logiche**
  - Quali sono e come si progettano le macchine elementari che costituiscono un calcolatore
- **Architettura ed organizzazione dei calcolatori elettronici**
  - Quali sono i componenti di un calcolatore, come sono interconnessi, come interagiscono
- **Linguaggio del processore**
  - Le istruzioni del processore, la programmazione in linguaggi assembly, corrispondenza tra linguaggi ad alto livello e linguaggio macchina



- Lezioni frontali ed esercitazioni in aula
- Esercitazioni guidate con l'ausilio di tutor didattici



Le attività didattiche utilizzeranno le piattaforme telematiche indicate dall'Ateneo:

- Microsoft Teams
  - Teams del corso (iscrizione tramite codice pubblicato sul sito docenti)
- Piattaforma Moodle per test, esercizi ed altro
  - <https://mooduni.unina.it/course/view.php?id=579>
  - Autenticarsi tramite OpenID Connect con le proprie credenziali UNINA



- Sito ufficiale del corso
  - <http://wpage.unina.it/rcanonic/didattica/ce1>
- Sito ufficiale del docente
  - <http://www.docenti.unina.it/roberto.canonico>
  - **IMPORTANTE:**  
Effettuare l'iscrizione al corso "Calcolatori Elettronici" selezionando l'opzione "Iscrizione alla mailing list" per ricevere comunicazioni dal docente via e-mail
  - Avvisi in bacheca, calendario esami, orario di ricevimento



Architettura dei Calcolatori  
Conte, Mazzeo, Mazzocca, Prinetto  
CittaStudi, 2015



Reti Logiche  
Bolchini, Brandolese, Salice, Sciuto  
Apogeo, 2008



Sistemi digitali e architettura dei calcolatori  
Progettare con tecnologia ARM  
D.M. Harris, S.L. Harris  
Zanichelli, 2017



Reti Logiche:  
complementi ed esercizi  
Fadini, Mazzocca  
Liguori, 1995

## Appunti sul processore Motorola MC68000

Set di istruzioni,  
modi di indirizzamento,  
un linguaggio assembly

A cura del prof. Bruno Fadini

2006

© Università degli Studi di Napoli Federico II  
Dipartimento di Informatica e Sistemistica

Revisione 19/4/2000

Appunti sul processore  
Motorola 68000  
Bruno Fadini, 2006

## The 68000's Instruction Set

We have included this appendix to save you the task of having to turn to secondary material when writing 68000 assembly language programs. Since most programmers are not interested in the encoding of instructions, details of instruction encoding have been omitted (i.e., the actual op-code bit patterns). Applications of some of the instructions have been provided to demonstrate how they can be used in practice.

Instructions are listed by mnemonic in alphabetical order. The information provided about each instruction is: its assembler syntax, its attributes (i.e., whether it takes a byte, word, or longword operand), its description in words, the effect its execution has on the condition codes, and the addressing modes it may take. The effect of an instruction on the CCR is specified by the following codes:

- U The state of the bit is undefined (i.e., its value cannot be predicted)
- The bit remains unchanged by the execution of the instruction.
- The bit is set or cleared according to the outcome of the instruction.

Unless an addressing mode is implicit (e.g., NOP, RESET, RTI, etc.), the legal source and destination addressing modes are specified by their assembly language syntax. The following notation is used to describe the 68000's instruction set.

DA, A0	Data and address register direct.
(A0)	Address register indirect.
(A0)+, -(A0)	Address register indirect with post-incrementing or pre-decrementing.
(d,A0), (d,A0,31)	Address register indirect with displacement, and address register indirect with indexing and a displacement.
ABS.W, ABS.L	Absolute addressing with a 16-bit or a 32-bit address.
(d,PC), (d,PC,31)	Program counter relative addressing with a 16-bit offset, or with an 8-bit offset plus the contents of an index register.
imm	An immediate value (i.e., literal) which may be 16 or 32 bits, depending on the instruction.

Manuale  
"The 68000 Instruction Set"

Materiale disponibile sul sito web del corso:  
<http://wpage.unina.it/rcanonic/didattica/ce1>



In condizioni normali, l'esame si svolgerà nella seguente modalità:

- Prova scritta + Prova orale
- *Eventuale esonero dalla prova scritta in caso di esito positivo delle prove intracorso*
- Prenotazione *obbligatoria* tramite sistema SEGREPASS
- Verbalizzazione digitale

Presentarsi all'esame con:

- Documento di riconoscimento
- PIN SEGREPASS per la verbalizzazione digitale
- Carta, penna, calcolatrice
- Eventuali altri ausili indicati durante il corso

## Esempio di compito assegnato alla prova scritta

Corso di CALCOLATORI ELETTRONICI  
Prof. R. Canonico  
Corso di Laurea in INGEGNERIA INFORMATICA  
Corso di Laurea in INGEGNERIA DELL'AUTOMAZIONE  
A.A. 2012-2013

ESAME DEL 10 MARZO 2014

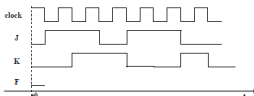
Alunno	
Cognome:	_____
Nome:	_____
Matricola:	____/____
Collocazione:	_____

### ESERCIZIO n.1

Progettare un contatore sincrono modulo 12, utilizzando flip-flop di tipo D.

### ESERCIZIO n.2

Un flip-flop F, di tipo JK "positive edge-triggered" (cioè attivo sul fronte di salita), sia sollecitato dai segnali J, K e clock il cui andamento temporale è di seguito riportato. Tracciare il corrispondente diagramma temporale per l'uscita F, assumendo  $F=0$  per  $t=0$ .



### ESERCIZIO n.3

Data la funzione booleana incompletamente specificata  $F(a,b,c,d)$  descritta dalla mappa di Karnaugh seguente:

cd \ ab	00	01	11	10
00	-	-	1	-
01	-	-	-	1
11	-	-	1	-
10	1	1	-	-

- Elencare tutti gli implicants primi della funzione che vale 1 sia nei punti in cui F vale 1, sia nei punti in cui F è non specificata, evidenziando quelli essenziali.
- Determinare tutte le espressioni equivalenti di F nella forma "somma di prodotti" e di costo minimo, usando il metodo della matrice di copertura.
- Trasformare in forma NAND una delle forme determinate al punto b).

### ESERCIZIO n.4

Illustrare le differenti tipologie di istruzioni di salto condizionato nel repertorio del processore 68000.

### ESERCIZIO n.5

Si scriva un programma assembly MC68000 che conti le occorrenze del carattere spazio (ASCII 32) presenti in una stringa allocata all'indirizzo  $S=\$8200$ . La stringa è terminata da '0' come in C/C++.



- Orario di ricevimento per il secondo semestre dell'a.a. 2022/23
  - martedì dalle 10.45 alle 12.45
    - presso il Dipartimento DIETI, via Claudio, palazzina 3/A stanza 4.18
    - oppure in modalità telematica tramite piattaforma Microsoft Teams sul canale di codice u07f5u3
- Email: [roberto.canonico@unina.it](mailto:roberto.canonico@unina.it)