

Storia dei Sistemi di Calcolo

Tutto sommato, potremmo accontentarci di una macchina con registri in quantità e i cui programmi sono fatti solo di 3 istruzioni:

- *Azzerare il contenuto di un registro*
- *Incrementare di uno il valore di un registro*
- *Saltare all'istruzione specificata se due registri sono uguali*

Come vedremo, non ci siamo accontentati...

Sistemi di numerazione

- Un sistema di calcolo, manuale, meccanico, elettrico o elettronico che sia, manipola dei simboli seguendo determinate regole.
- I primi simboli per cui è nata l'esigenza di una loro manipolazione sono naturalmente i numeri, utili ad esempio (ma non solo) per indicare quantità ben precise.
- Come ha fatto notare Bertrand Russell (matematico e filosofo del ‘900):
 - *Possono essere state necessarie [all'uomo] ere intere per realizzare che una coppia di contadini e un paio di giorni erano entrambi istanze del numero 2*

Sistemi di numerazione

- Possiamo quindi affermare con certezza che l'invenzione dei numeri sia stata uno dei grandi passi in avanti dell'umanità (ovviamente a sua volta preceduto dall'invenzione del linguaggio)
- Ma l'invenzione dei numeri fu seguita da un altro grande passo: la messa a punto di sistemi per rappresentarli; sistemi che vennero sviluppati addirittura prima della scrittura
- In ritrovamenti risalenti a circa 10.000 anni fa troviamo bastoncini, nodi, gettoni usati nella modalità di rappresentazione più semplice, quella **unaria**, in cui per indicare ad esempio il numero 5 si usano 5 bastoncini.

Sistemi di numerazione

- Naturalmente, la rappresentazione unaria è piuttosto scomoda per numeri anche piccoli.
- Intorno al 3000 AC si sviluppano in Egitto i primi sistemi di numerazione addizionali, in cui cioè il numero rappresentato è scritto usando simboli di valore diverso, che vanno sommati fra loro.



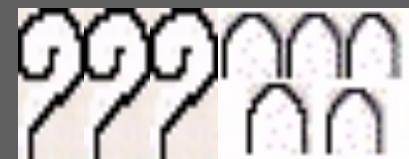
- Ecco ad esempio il numero 350, poiché ogni arco ha valore 10 e ognuno dei 3 simboli successivi vale 100.

Sistemi di numerazione

- Notate che non si tratta di un sistema posizionale, perché il valore di ogni simbolo non dipende dalla sua posizione, e quindi:



è uguale a



- Si sviluppano anche sistemi sia additivi che sottrattivi, il più famoso di tutti sicuramente quello romano, il quale usa un criterio “posizionale relativo”, in quanto la posizione di una cifra rispetto ad un’altra stabilisce se la prima vada sommata o sottratta alla seconda:
- Così: $\text{XI} = 10 + 1 = 11$, mentre $\text{IX} = 10 - 1 = 9$

Sistemi di numerazione

- È intorno al 500 DC che in India viene messo a punto il sistema posizionale addittivo in base 10 che usiamo ancora oggi, e nel quale l'uso dello “zero” gioca un ruolo fondamentale.
- In realtà i babilonesi avevano molto tempo prima sviluppato un sistema analogo, ma in base 60 e senza l'uso dello zero.
- L'uso di 10 simboli e di uno zero esplicito rende il sistema di numerazione indiano molto più semplice, sintetico e versatile di quello babilonese (in cui 3 soli simboli erano combinati mediante una tecnica addittiva per formare le 59 cifre, e lo zero non era una cifra ma una “posizione vuota”).
- Il sistema indiano si diffonde velocemente verso il mondo arabo.
- “zero” deriva dall’arabo “*sifr*”, mentre in sanscrito era indicato con la parola “*sunya*”, che significa *vuoto*.

Sistemi di numerazione

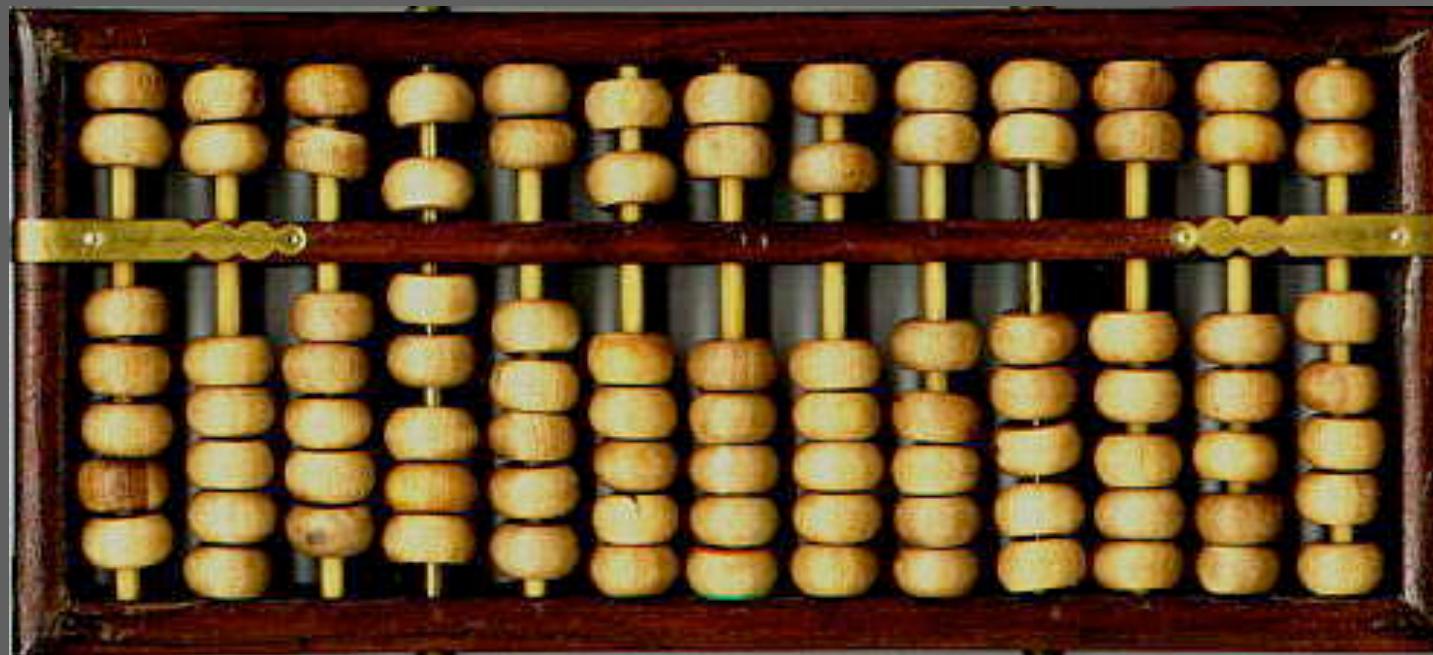
- Intorno al 820 DC **Mohammed ibn-Musa al-Khuwarizmi** scrive alcuni trattati di algebra e aritmetica descrivendo e impiegando largamente il sistema di numerazione indiano.
- È a partire da questi trattati che si diffonderà in Europa il sistema di numerazione arabo-indiano: posizionale addittivo in base 10.
- La diffusione in Europa si deve in particolare a **Leonardo Pisano**, più noto come il **Fibonacci** (da *Filius Bonacci*, era infatti figlio di Guglielmo dei Bonacci, ricco mercante pisano)
- Nel suo **Liber Abaci**, del 1202, Fibonacci introduce il sistema arabo-indiano, e i metodi che usiamo ancora oggi per eseguire a mano le quattro operazioni e per estrarre la radice quadrata.

Supporti meccanici al calcolo

- Molto presto nasce l'esigenza di un aiuto per facilitare l'esecuzione dei calcoli.
- L'**abaco** è sicuramente il più noto e antico degli strumenti usati, già usato in Cina millenni fa, se ne trova una descrizione in un libro circa 200 anni AC.
- Diffuso in tutto il mondo, è ancora usato in alcuni paesi.
- Non è però un vero e proprio strumento di calcolo, ma piuttosto uno strumento per memorizzare i risultati intermedi dei calcoli.

Supporti meccanici al calcolo

- Un abaco cinese. Ogni guida rappresenta un peso diverso. Ad esempio: unità, decine, centinaia, migliaia, che vengono contate dalle pietre della guida corrispondente.



Supporti meccanici al calcolo

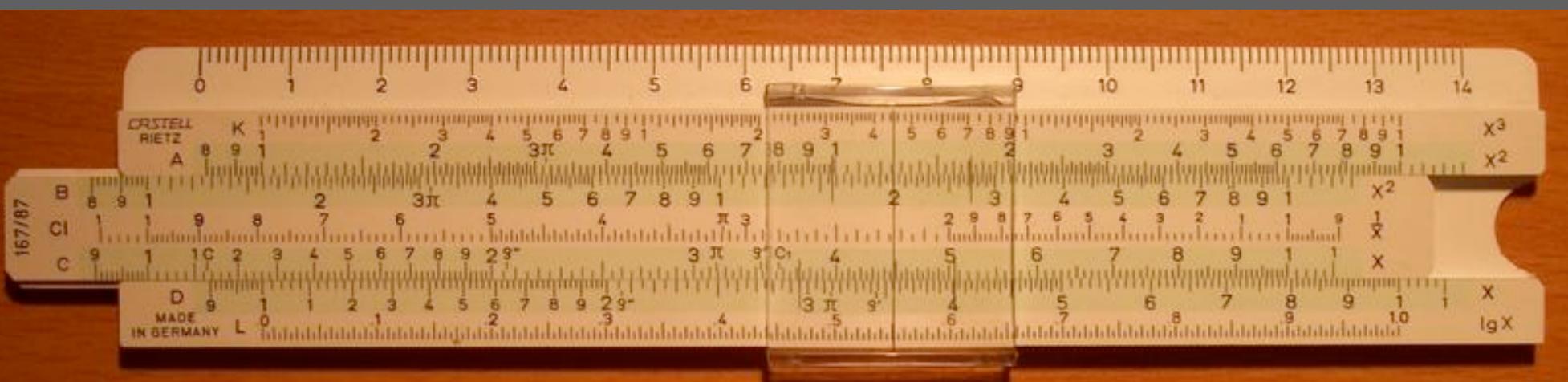
- Il ‘600 è un secolo chiave per lo sviluppo della matematica applicata:
- Nell’industria bellica si sviluppano i metodi balistici, e quindi nasce l’esigenza di calcolare la traiettoria dei proiettili sparati dai cannoni.
- Lo sviluppo della navigazione transoceanica richiede tavole matematiche necessarie per la navigazione.
- L’esigenza di un aiuto al calcolo mediante supporti meccanici diviene sempre più pressante.
- A fine ‘500 incomincia a diffondersi l’uso della virgola/punto per indicare i numeri decimali, e l’uso dei numeri negativi.

Supporti meccanici al calcolo

- All'inizio del '600 (circa 1614) lo scozzese **John Napier** (più noto in Italia come **Nepero**) inventa i **logaritmi**.
- Dal punto di vista del calcolo automatico, l'importanza dei logaritmi sta nel fatto che permettono di trasformare una operazione di moltiplicazione in una di addizione: $\log(a * b) = (\log a) + (\log b)$
- E quindi, per calcolare “ $a * b$ ”:
 - Cerchiamo in una tavola di logaritmi $(\log a)$ e $(\log b)$
 - Sommiamo $(\log a)$ e $(\log b)$ ottenendo c
 - Cerchiamo nella tavola dei logaritmi quel numero il cui logaritmo è c . Il numero trovato è uguale ad “ $a * b$ ”

Supporti meccanici al calcolo

- L'inglese **Edmund Gunter** sfrutta questa proprietà dei logaritmi per inventare intorno al 1620 il **regolo calcolatore**: due righelli scorrevoli graduati con scale logaritmiche che permettono di eseguire meccanicamente moltiplicazioni e divisioni.
- Perfezionato nei secoli, il regolo rimane largamente in uso fino agli anni '70 del secolo XX, ossia fino alla diffusione su grande scala delle calcolatrici elettroniche portatili.
- Un esempio di regolo calcolatore moderno:



Supporti meccanici al calcolo

- Nel 1623 il tedesco **Wilhelm Schikard**, esperto meccanico, inventa la prima vera e propria macchina calcolatrice meccanica: **l'orologio calcolatore**.
- L’”orologio” poteva eseguire somme e sottrazioni mediante ruote dentate collegate ad un visualizzatore numerico
- Era in grado di gestire i riporti delle somme, e della macchina ne fu realizzato un solo prototipo in legno andato distrutto in un incendio.
- Si ha notizia di tale invenzione solo grazie ad una corrispondenza epistolare tra Schikard e il famoso astronomo **Johannes von Kepler** (Giovanni Keplero)

Supporti meccanici al calcolo

- La più famosa macchina per calcoli meccanici, la **Pascalina**, fu progettata, nel 1641, a 18 anni, dal francese **Blaise Pascal** (il famoso filosofo), per aiutare il padre esattore delle tasse.
- La macchina poteva eseguire solo addizioni, mentre le sottrazioni potevano essere eseguite col metodo del “complemento a 10”
- Fu costruita nel 1645 in 50 esemplari



Supporti meccanici al calcolo

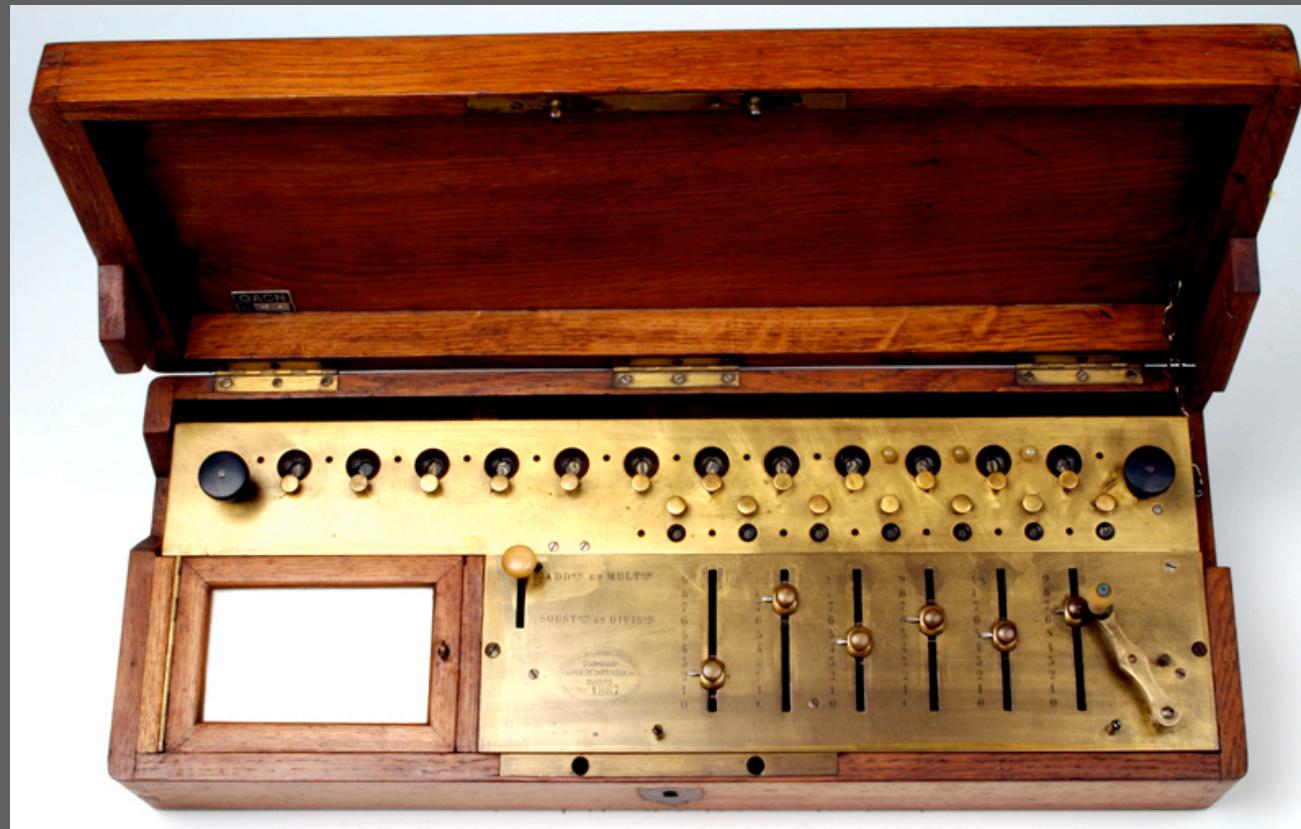
- Tra il 1672 e il 1673 il tedesco **Wilhelm Leibniz** progetta la prima macchina calcolatrice in grado di eseguire moltiplicazioni e divisioni (mediante somme e sottrazioni ripetute). Costruita solo nel 1694 non funzionò mai molto bene, a causa dei limiti tecnologici del tempo.
- In campo informatico, ricordiamo Leibniz anche per altri contributi:
 1. Concepì il **sistema di numerazione binario** (pare addirittura ispirato dall'I Ching, un antico sistema di divinazione cinese a due simboli)
 2. Fu un precursore della **moderna logica matematica**, ipotizzando che il ragionamento umano potesse essere riscritto in formule e manipolato con regole computazionali come si fa con i numeri
- Scrisse tra l'altro: *It is unworthy of excellent men to lose hours like slaves in the labor of calculation, which could be safely relegated to anyone else if machines were used”*

Supporti meccanici al calcolo

- nel 1725 il francese **Basile Bouchon** ha l'idea di usare rotoli di carta perforata su telai per stabilire l'ornamento da riprodurre sulla stoffa.
- nel 1726 il suo collaboratore **Jean-Baptiste Falcon** migliora il progetto utilizzando sequenze di schede, rendendo più semplice il cambiamento di “programmazione” della macchina.
- nel 1801 **Joseph Jacquard** usa schede perforate metalliche per il controllo di un telaio, in gran parte automatico, noto *come telaio Jacquard* (ancora oggi nel campo del lavoro a maglia si usa l'espressione *lavoro a Jacquard*)
- Dal 1804 al 1814 verranno prodotti più di 10.000 telai Jacquard, e la sua invenzione rivoluzionerà la produzione tessile del XIX secolo.
- L'idea delle **schede perforate** rimane in uso per la programmazione dei moderni computer fino intorno al 1980 (ci torneremo più avanti)

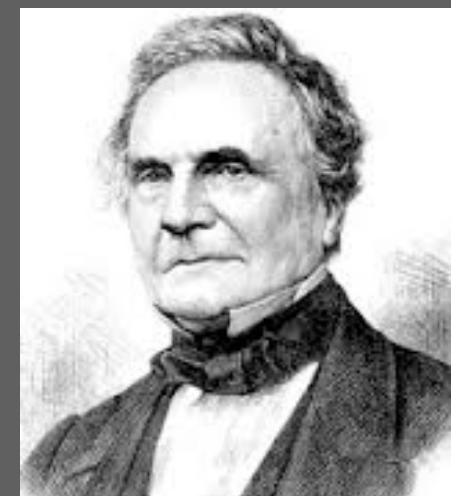
Supporti meccanici al calcolo

- Nel 1822 il francese **Thomas de Colmar** costruisce il primo prototipo dell'**aritmometro**, una macchina calcolatrice portatile in grado di eseguire le 4 operazioni con risultati fino a 12 cifre.
- Dal 1823 al 1878 ne furono prodotti più di 1500 esemplari



Supporti meccanici al calcolo: L'Analytical Engine di Charles Babbage

- **Charles Babbage** (1792 – 1871), inglese, fu una figura eminente del suo tempo.
- Gli fu assegnata la cattedra di Matematica a Cambridge nel 1828, La stessa che fu di Isaac Newton e, ai giorni nostri, di Stephen Hawking.
- Lasciò tuttavia la cattedra nel 1839 per dedicarsi a tempo pieno alle sue idee di inventore.
- Fu membro delle più importanti società scientifiche inglesi ed europee, e amico di Charles Darwin e Pierre Simon de Laplace.



L'Analytical Engine di Charles Babbage

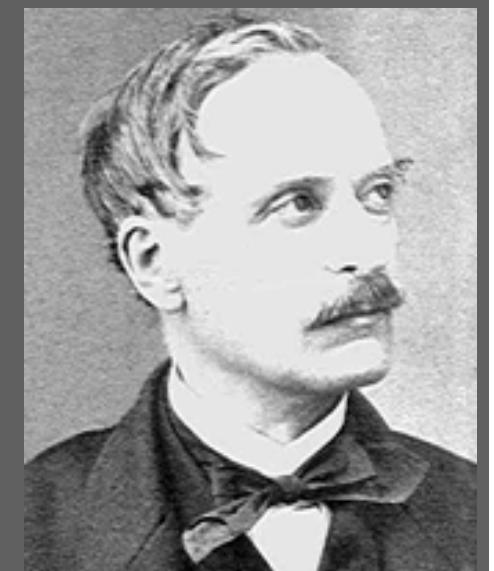
- Tra gli anni 1820 e 1830 Babbage concepisce e in parte realizza un dispositivo noto come **difference engine**, un dispositivo meccanico in grado di eseguire alcuni tipi di operazioni aritmetiche, anche su numeri molto grandi.
- Babbage ottiene anche cospicui finanziamenti pubblici per la realizzazione della sua invenzione, che tuttavia non vede mai completamente la luce.
- A parte i problemi tecnici, la ragione è che, a partire dai primi anni 30, Babbage si dedica ad una nuova invenzione: una macchina in grado di calcolare funzioni arbitrariamente complesse su un numero qualsiasi (ma comunque limitato) di variabili: l'**Analytical Engine**

L'Analytical Engine di Charles Babbage

- Da subito, Babbage concepisce l'idea di controllare il funzionamento dell'Analytical Engine attraverso schede perforate: l'intuizione gli deriva dall'idea del telaio di Jacquard.
- Babbage incomincia a chiedere finanziamenti per realizzare questa sua nuova invenzione, ma senza molto successo, visti i soldi già spesi e gli scarsi risultati ottenuti con il progetto precedente.
- Incomincia allora a cercare fuori dall'Inghilterra quell'interesse per le sue idee che non era stato in grado di sollevare in patria, e nell'agosto del 1840 viene invitato a Torino, a tenere una serie di conferenze al “**Secondo congresso degli Scienziati Italiani**” patrocinato da re Carlo Alberto di Savoia.
- Babbage lascerà poi alcuni suoi disegni alla città, che sono conservati all'**Accademia delle Scienze di Torino**

L'Analytical Engine di Charles Babbage

- Nel partecipare al congresso, l'intenzione di Babbage è di sollevare l'interesse di **Giovanni Plana**, astronomo torinese, professore dell'Università di Torino, tra gli scienziati più noti e influenti a livello europeo in quel periodo (Plana fu tra l'altro il fondatore dell'Osservatorio Astronomico di Torino)
- Babbage spera che Plana voglia scrivere e pubblicare un resoconto delle sue conferenze, incarico che però Plana affida ad un suo allora sconosciuto allievo, **Luigi Federico (conte di) Menabrea**, ingegnere del genio militare.
- (Menabrea diverrà poi generale di Garibaldi, ministro del regno e primo ministro dal 1867 al 1869. Il suo interesse per la scienza rimase comunque sempre vivo, e fu tra l'altro professore di Scienza delle Costruzioni all'Università di Torino dal 1846 al 1860)

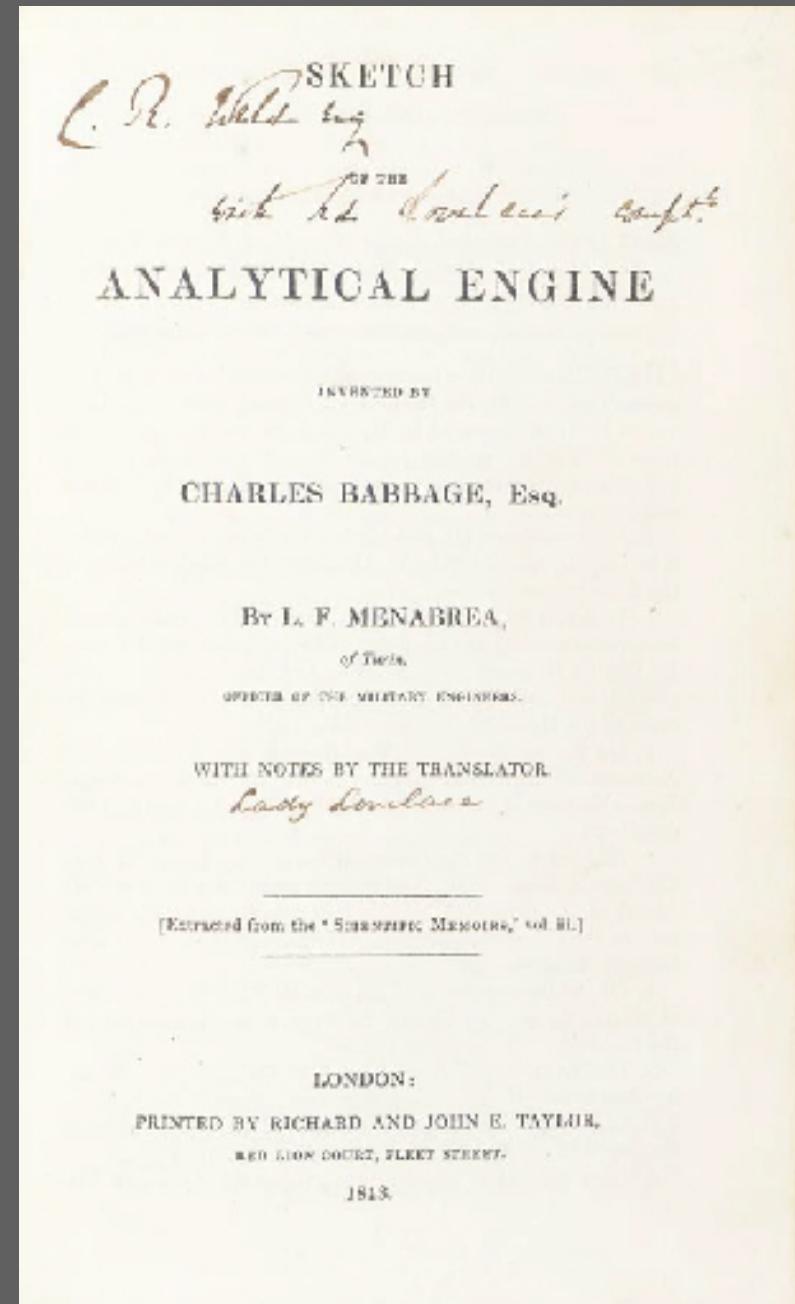


L'Analytical Engine di Charles Babbage

- Menabrea scrive un articolo che riassume le idee dell'Analytical Engine, e l'articolo viene pubblicato, in francese, nell'ottobre del 1842 sul n. 82 della rivista della *Bibliothèque Universelle de Genève*.
- All'inizio del 1843, **Ada Augusta Byron, contessa di Lovelace** (e figlia del poeta George Byron) su consiglio di Charles Wheatstone – (amico di famiglia e co-inventore del telegrafo) traduce il lavoro di Menabrea per il Taylor's *Scientific Memoirs*, una rivista inglese specializzata nella pubblicazione di articoli scientifici e resoconti di congressi esteri.
- Ada Lovelace invia la traduzione a Babbage, il quale la esorta ad aggiungere note di chiarimento (che alla fine risulteranno più lunghe dell'articolo originale) al lavoro di Menabrea.

L'Analytical Engine di Charles Babbage

- L'articolo di Menabrea, tradotto e corredata delle ampie note di Ada viene pubblicato nel settembre del 1843, ed è tradizione considerarlo il primo articolo nella storia dell'informatica.
- In figura, la copertina della rivista con l'articolo di Menabrea. Il nome della traduttrice aggiunto a mano.
- Copia dell'originale è stata battuta all'asta da Christies nel 2013 per 170.000 \$.



L'Analytical Engine di Charles Babbage

- L'articolo descrive molti aspetti simili all'architettura dei moderni computer e alla loro programmazione quasi cento anni prima che queste idee fossero “riscoperte”, nel XX secolo.
- Con un po' di esagerazione, è ormai tradizione considerare Charles Babbage l'inventore del primo computer moderno, e Ada Lovelace la prima programmatrice nella storia dell'informatica.
- Sfortunatamente, le idee di Babbage e di Ada Lovelace passarono sostanzialmente sotto silenzio, e furono poi dimenticate per lungo tempo...



L'Analytical Engine di Charles Babbage

- L'Analytical Engine era sostanzialmente un computer meccanico a schede perforate: come abbiamo visto, l'idea delle schede proveniva dal telaio per tessitura inventato da Jacquard: una scheda perforata permetteva di stabilire quale combinazione di fili di diverso colore dovessero essere usati in ogni riga di un tessuto.
- Ma nel telaio di Jacquard era necessaria una scheda per ogni riga del tessuto che doveva essere prodotto (a quei tempi esisteva un ritratto di Jacquard fatto al telaio usando 24.000 schede).
- Un'intuizione fondamentale nell'Analytical Engine fu l'idea di usare *cicli* per i calcoli che dovevano essere ripetuti più volte, in questo modo usando un numero finito e limitato di schede per specificare calcoli che potevano essere ripetuti un numero indefinito di volte.

L'Analytical Engine di Charles Babbage

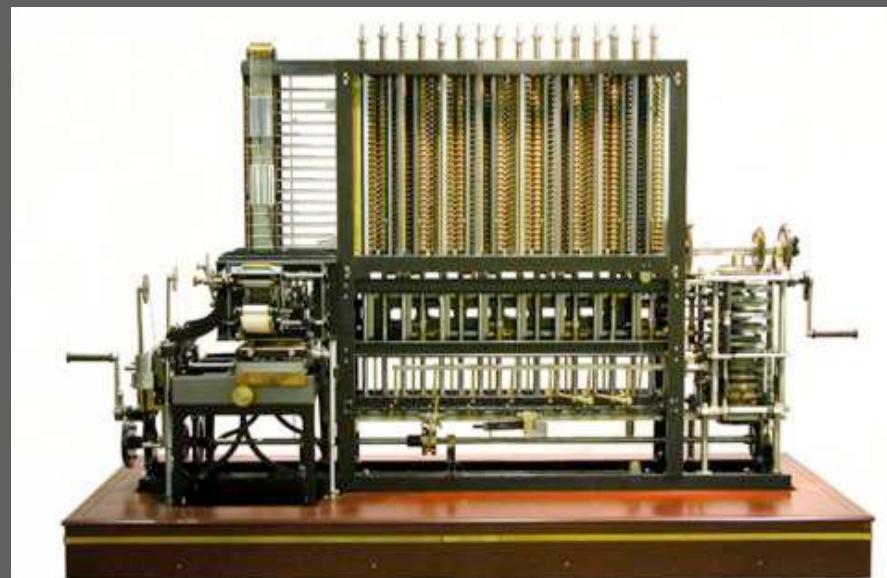
- La macchina di Babbage era progettata per eseguire delle **Operazioni** su delle *Variabili*.
- Le operazioni venivano eseguite da un complesso dispositivo chiamato **mill** (mulino, macinino) guidato dalle schede perforate.
- Il *mill* operava sulle *Variabili*: cilindri di ottone su cui erano impilati più dischi numerati da 0 a 9, in grado così di rappresentare numeri positivi e negativi in notazione decimale.
- Le *Variabili* potevano memorizzare i valori di input di un calcolo da eseguire, il risultato di computazioni intermedie da usare poi in altre computazioni, e naturalmente il risultato finale delle computazioni: schede speciali permettevano di specificare quali *Variabili* dovevano memorizzare quali valori (iniziali, intermedi e finali)

L'Analytical Engine di Charles Babbage

- Nell'Analytical Engine emergono quindi già in modo abbastanza chiaro alcuni concetti di base dei moderni processori:
- **L'unità di controllo**, il Mill, che opera sulle *Variabili*.
- Il **programma** che deve essere eseguito dal Mill, rappresentato da una serie di schede perforate.
- Il **concetto di memoria**, usata per contenere i dati iniziali, intermedi e finali del calcolo, rappresentato dalle *Variabili* della macchina (in sostanza, i registri di una moderna CPU).
- L'idea di **computazione parallela**, poiché diverse operazioni intermedie venivano eseguite in parallelo su diverse *Variabili*.
- Il concetto di **input e output**, rappresentato da quelle *Variabili* che contenevano i valori di partenza di una computazione, e dalle *Variabili* designate per memorizzare i risultati finali.

L'Analytical Engine di Charles Babbage

- chi vuole approfondire questo argomento può trovare in rete molte informazioni. Un buon punto di partenza è il seguente:
 - www.fourmilab.ch/babbage/contents.html
- Qui si trova tra l'altro l'articolo di Menabrea tradotto da Ada Lovelace, biografie dei protagonisti di questa storia, riproduzioni dei disegni dei progetti di Babbage, e alcuni programmi Java che permettono di emulare il funzionamento dell'Analytical Engine.
- In figura una modello funzionante dell'Analytical Engine, costruito sulla base dei progetti di Babbage. In alto, verticali, i cilindri che implementano le *Variabili*.



L'algebra di Boole

- Nel 1854 l'inglese **George Boole** pubblica un lavoro scientifico dal titolo:

An Investigation of the Laws of Thought, on Which Are Founded the Mathematical Theories of Logic and Probabilities

- Dove vengono illustrati i principi dell'**Algebra Booleana**, lo strumento concettuale su cui si basano tutti i calcolatori digitali moderni.
- In realtà, al tempo fu considerato un lavoro puramente teorico, fino a che nella seconda parte degli anni '30 del XX secolo molti ricercatori mostrarono come era possibile costruire circuiti digitali che funzionavano usando l'algebra booleana (ci torneremo più avanti).

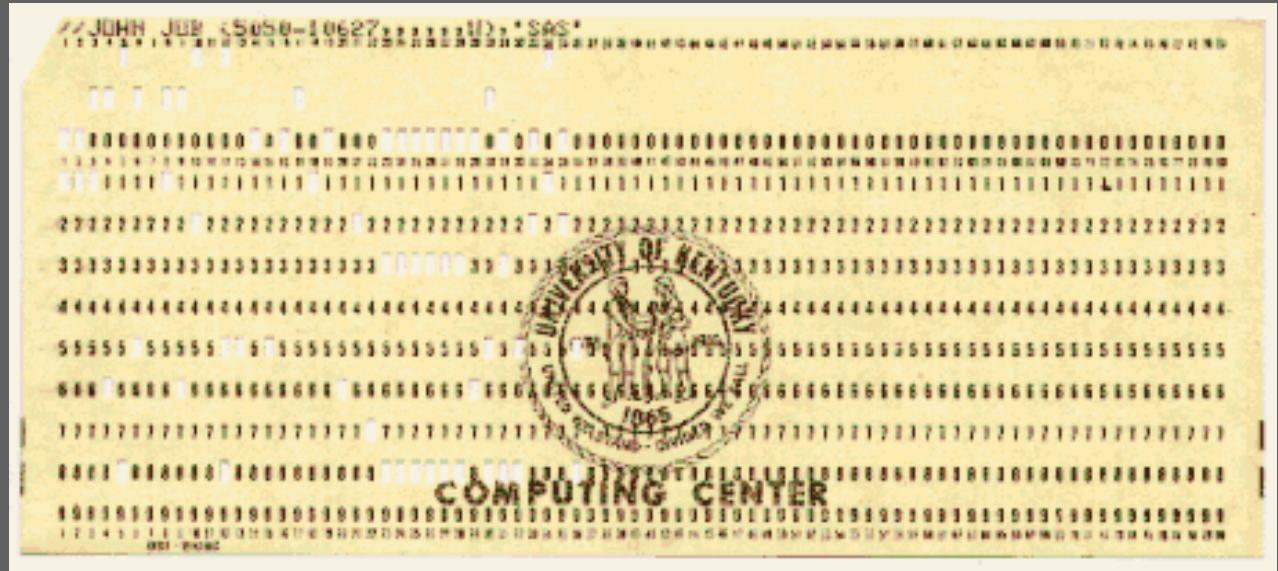
Supporti elettro-meccanici al calcolo

- In occasione del censimento del 1890, negli USA viene bandita una gara per un sistema di conteggio in grado di elaborare velocemente i dati raccolti dal censimento (non erano infatti ancora stati completamente elaborati i dati del censimento precedente, gestiti completamente a mano...)
- **Herman Hollerith** vince il concorso con il suo sistema **Elettrico di Tabulazione**.
- Si trattava di una macchina elettro-meccanica in grado di gestire schede perforate, ognuna delle quali conteneva le informazioni anagrafiche di un cittadino.
- In sostanza quindi le schede costituivano l'input della macchina (nei decenni successivi, nei computer del XX secolo, verranno usate per contenere anche il programma da eseguire)

Supporti elettro-meccanici al calcolo

- Vediamo brevemente il funzionamento della macchina di Hollerith.
- Una scheda perforata era un cartoncino della dimensione più o meno di un dollaro, e rappresentava l'informazione attraverso la presenza o l'assenza di fori in posizioni predefinite della scheda.
- Ad esempio, in una certa posizione stabilita a priori, la presenza di un foro indicava “sposato”, la sua assenza indicava “nubile/celibe”.

- Una scheda perforata Hollerith



Supporti elettro-meccanici al calcolo

- Le risposte al censimento fornite da ciascun cittadino venivano codificate da un operatore/operatrice su una scheda mediante un pantografo meccanico.
- Ogni scheda era suddivisa in 288 zone che rappresentavano le varie informazioni anagrafiche, e ogni zona veniva forata o meno attraverso il pantografo in base alle risposte del cittadino associato a quella scheda.



Supporti elettro-meccanici al calcolo

- Una volta codificate le informazioni sulle schede, queste potevano essere raggruppate in pacchetti ed elaborate automaticamente dal tabulatore.
- ogni scheda veniva inserita in un dispositivo con una batteria di aghi retrattili, che in assenza di perforazione venivano fermati dal cartoncino. Altrimenti, l'ago finiva in una vaschetta piena di mercurio, chiudendo un circuito elettrico che faceva avanzare di uno scatto uno di 40 contatori necessari a contare le diverse risposte.
- Ogni contatore veniva incrementato quando si verificava una certa combinazione di informazioni che si voleva registrare (ad esempio, *persona sposata senza figli*)

Supporti elettro-meccanici al calcolo

- Tabulator con 40 contatori, pantografo e selezionatore delle schede



Supporti elettro-meccanici al calcolo

- la macchina poteva esaminare fino a 800 schede al minuto, una velocità eccezionale per quei tempi visto che si era abituati e elaborare tutto manualmente.
- Il lavoro di perforazione e tabulazione dei dati anagrafici del censimento americano del 1890 (63 milioni di persone divise in circa 150.000 comunità) durò un solo mese!
- Hollerith in seguito perfezionò la sua macchina aggiungendo la possibilità di sommare i numeri codificati sulle schede.
- Questo ne estese il campo di utilizzo, ad esempio fu usata dalle compagnie ferroviarie, dalle assicurazioni, e così via

Supporti elettro-meccanici al calcolo

- Nel 1896 Hollerith fonda la **Tabulating Machine Corporation** per la produzione e commercializzazione della sua invenzione.
- Nel 1924, dalle ceneri di questa compagnia nascerà la **International Business Machine Corporation**, più nota come **IBM**
- La IBM dominò il mercato dei computer del ventesimo secolo fino più o meno agli anni '70: la maggior parte dei computer e degli altri dispositivi elettronici ed elettromeccanici necessari per sviluppare i programmi erano prodotti dalla IBM.
- Nota di colore: negli anni '60, per indicare il dominio della IBM, era stata coniata l'espressione “**IBM and the seven dwarfs**”, dove i sette nani erano la Digital Equipment, la Control Data, la General Electric, la RCA, la Univac, la Burroughs, e la Honeywell

Le schede perforate

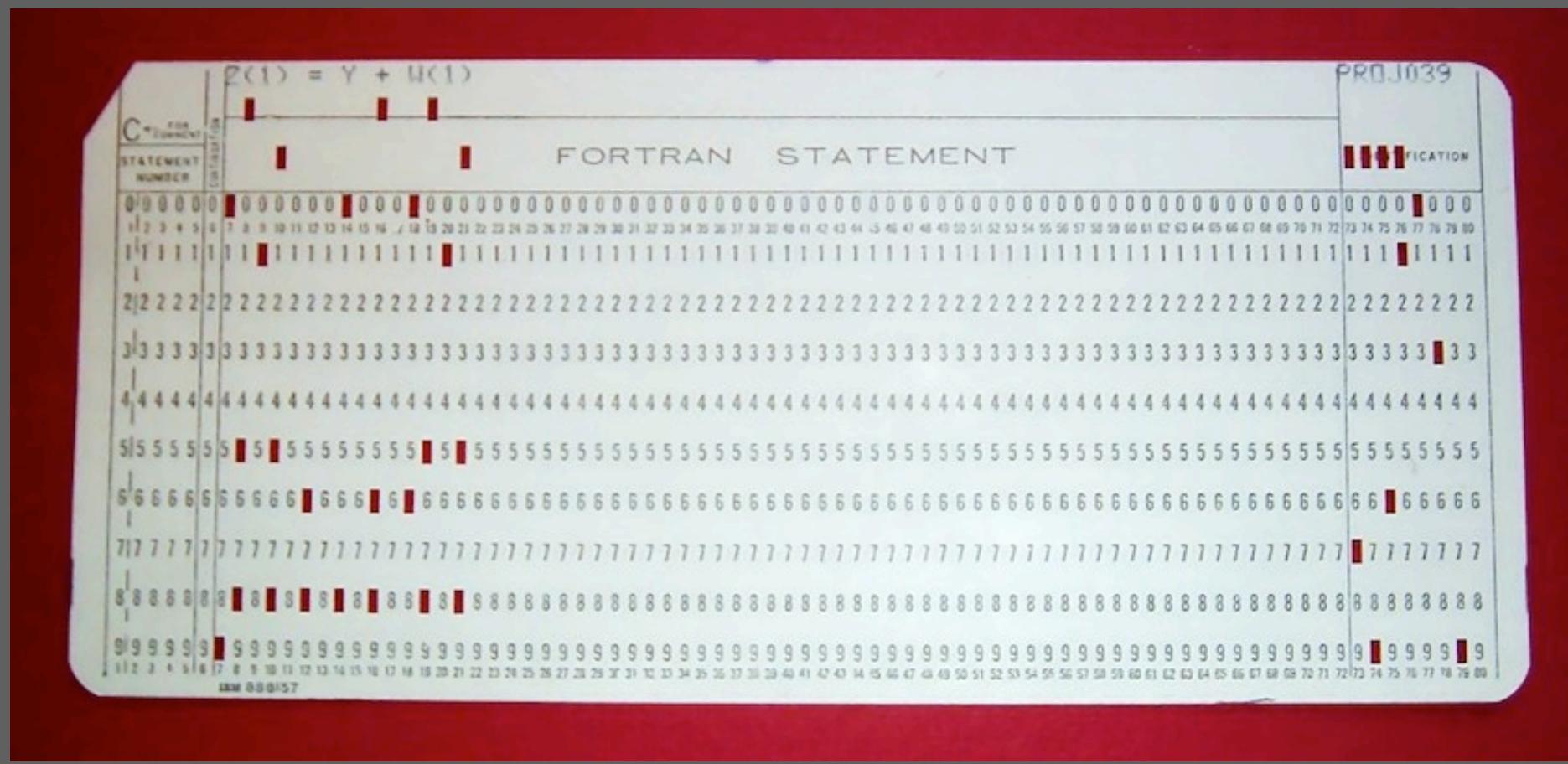
- Le schede perforate ricopriranno un ruolo importante nell'uso dei computer per buona parte del XX secolo.
- Le schede rappresentavano un metodo semplice ed economico per codificare informazioni che dovevano essere elaborate in modo automatico.
- Col tempo, perforatori di schede e lettori di schede furono modificati in modo da essere sempre più veloci, e da poter inserire sulle schede non solo i dati da manipolare, ma anche i programmi da eseguire al computer.
- Era normale negli anni '60 e '70 vedere i programmatori andare in giro con un pacchetto di schede su cui era memorizzato il programma da far girare sul computer del centro di calcolo.

Le schede perforate

- In sostanza, quasi tutti i programmatori facevano girare i loro programmi codificandoli in un pacchetto di schede perforate.
- Le schede potevano essere perforate usando un opportuno **perforatore di schede**, che era simile, in dimensioni ed aspetto, a una macchina da scrivere.
- Ogni scheda conteneva una linea del programma da eseguire. Altre schede contenevano ciascuna un dato di input del programma.
- Una macchina stampante era in grado di leggere il pacchetto di schede che costituiva un programma con i suoi dati e di stampare il programma su carta a modulo continuo.
- Questo permetteva di analizzare il programma prima di farlo girare, correggendo eventuali schede sbagliate.

Le schede perforate

- Una scheda perforata IBM con una linea di codice in FORTRAN:
 $Z(1) = Y + W(1)$ (effettivamente codificata nei fori praticati sulla scheda)



Le schede perforate

- Studenti negli anni '60 e '70. le macchine di fronte a loro sono perforatori di schede. Notate pacchetti di schede ancora da perforare sulle mensole a destra



Le schede perforate

- I programmatori professionisti potevano anche contare su personale preposto alla trasposizione su schede perforate dei loro programmi



Le schede perforate

- Carta per stampare programmi a modulo continuo e stampante IBM a modulo continuo



Le schede perforate

- Per far girare un programma era necessario consegnare il pacchetto di schede che lo codificava ad un operatore del centro di calcolo.
- Il computer su cui venivano fatti girare i programmi aveva le dimensioni di qualche metro cubo ed era collegato direttamente ad un lettore di schede oppure ad un lettore di nastro magnetico a sua volta collegato ad un lettore di schede (anche questi della dimensione di qualche metro cubo).
- Più pacchetti di schede (ossia i programmi da far girare sul computer) venivano inseriti nel lettore di schede, che era in grado di ospitare normalmente un gran numero di schede, e quindi più programmi da far girare, uno dopo l'altro.
- Una scheda senza angolo sinistro tagliato veniva inserita dall'operatore tra un pacchetto di schede e l'altro, per distinguere appunto i vari “programmi”.

Le schede perforate

- Pacchetti di “programmi”. Notate le scritte a penna che indicano il programma codificato in un certo pacchetto di schede, oppure il proprietario (ad esempio la matricola) di quel particolare pacchetto.



Le schede perforate

- Un'operatrice inserisce pacchetti di schede nel lettore di schede.



Le schede perforate

- Solo quando un numero sufficiente di pacchetti di schede era stato raccolto ed elaborato dal lettore di schede, i programmi potevano essere effettivamente mandati in esecuzione.
- Era quindi normale dover aspettare alcune ore (ma anche il giorno successivo, o più giorni) per avere i risultati di una esecuzione
- L'operatore si incaricava di stampare i risultati di ogni programma eseguito e di consegnare la stampa, insieme con il pacchetto di schede relativo, al suo proprietario.
- Ovviamente c'erano varianti a questo modo di procedere, in base al tipo di applicazioni, al tipo di computer e al tipo di programmatore coinvolti. Ma il protocollo generale era questo (ci torneremo più in dettaglio in occasione della storia dei sistemi operativi).

I calcolatori analogici

- Tra la fine dell'800 e la prima parte del '900, vengono progettati e costruiti alcuni esempi di **calcolatori analogici**, in grado di risolvere specifici problemi.
- A differenza di un calcolatore digitale, che lavora con grandezze discrete (tipicamente numeri), i calcolatori analogici lavorano con grandezze reali, che variano in modo continuo.
- Ad esempio, una grandezza reale può essere rappresentata da un livello di tensione, che può assumere qualsiasi valore all'interno di un intervallo prestabilito.
- Sono stati costruiti calcolatori analogici che lavorano su diverse grandezze fisiche: oltre a quelle elettriche (tensione, corrente) sono state usate ad esempio grandezze idrauliche (livello di un liquido) e meccaniche (posizione di un ingranaggio)

I calcolatori analogici

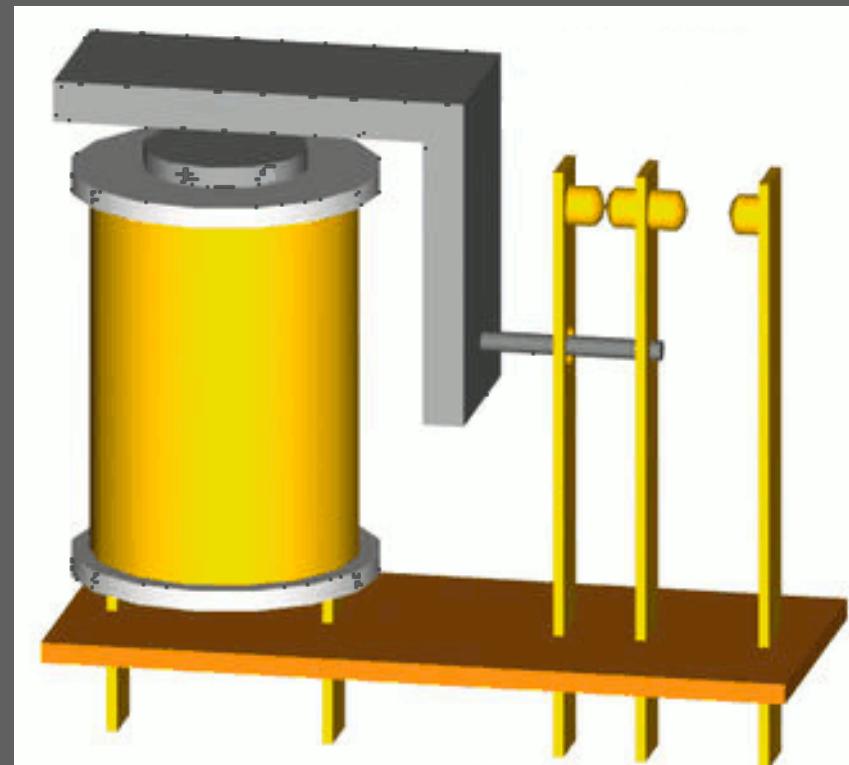
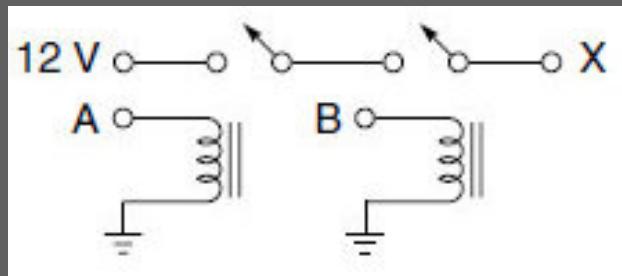
- Il primo calcolatore analogico generale, in grado di risolvere equazioni differenziali del terzo ordine, fu progettato nel 1931 da **Vannevar Bush**.
- Bush è considerato il precursore degli ipertesti. In un articolo del 1945, “**As we may think**”, egli ipotizza una apparecchiatura, il **memex**, con cui si possano raccogliere e organizzare testi, e predice la comparsa di “*nuovi tipi di encyclopedie confezionate con una rete di percorsi associativi che le collegano, pronte ad essere inserite nel memex e qui ampliate*”.
- I calcolatori analogici erano tuttavia molto difficili da programmare: si trattava di riconfigurarne in modo complesso e sofisticato i collegamenti elettrici di cui erano fatti per ogni programma.
- Inoltre avevano una capacità di rappresentare diversi tipi di problemi assai poco flessibile. In particolare, era difficile rappresentare (e poi risolvere) problemi che non coinvolgevano grandezze reali.

La nascita dei computer digitali

- Per una serie di eventi concomitanti, il periodo 1930 - 1945 risulta fondamentale per il concepimento, la progettazione e lo sviluppo dei primi computer digitali “moderni”, di solito costruiti grazie all’ingegno, all’intuito, agli sforzi personali e spesso un pizzico di pazzia da singoli ricercatori e inventori.
- Un ruolo importante lo ebbero alcune scoperte e avanzamenti tecnologici degli anni precedenti, in particolare l’invenzione dei **relè** e delle **valvole termoioniche**.
- In modo spesso indipendente si verificò anche un sensibile avanzamento e sviluppo delle idee collegate al calcolo e all’elaborazione automatica di informazioni.
- Le tensioni internazionali e la II guerra mondiale costituirono un enorme impulso, e produssero un cospicuo investimento di sforzi e risorse economiche in nuove ricerche scientifiche e tecnologiche.⁴⁹

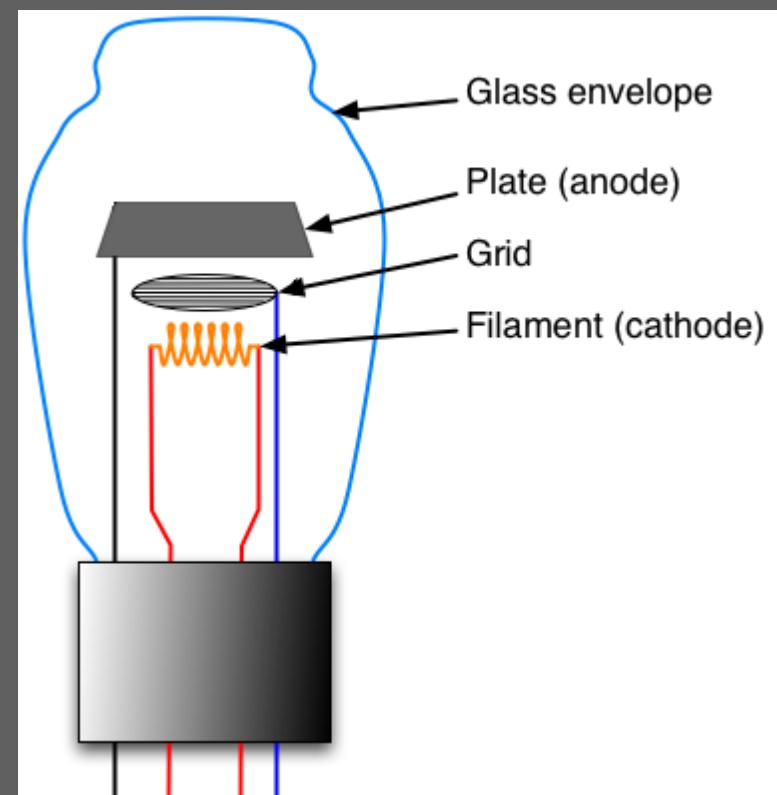
La nascita dei computer digitali

- Il funzionamento di un qualsiasi circuito digitale (e un computer digitale è sostanzialmente un circuito digitale molto grande) si basa sull'uso di **interruttori a comando elettrico**.
- Un tale tipo di interruttore permette o interrompe un flusso di corrente elettrica tramite un segnale elettrico.
- Il primo esempio di questo tipo di interruttore è il **relè**, inventato nel 1835 dall'americano **Joseph Enry**, in cui una elettrocalamita fa scattare un interruttore.
- Ecco una porta AND ottenuta collegando insieme due relè:



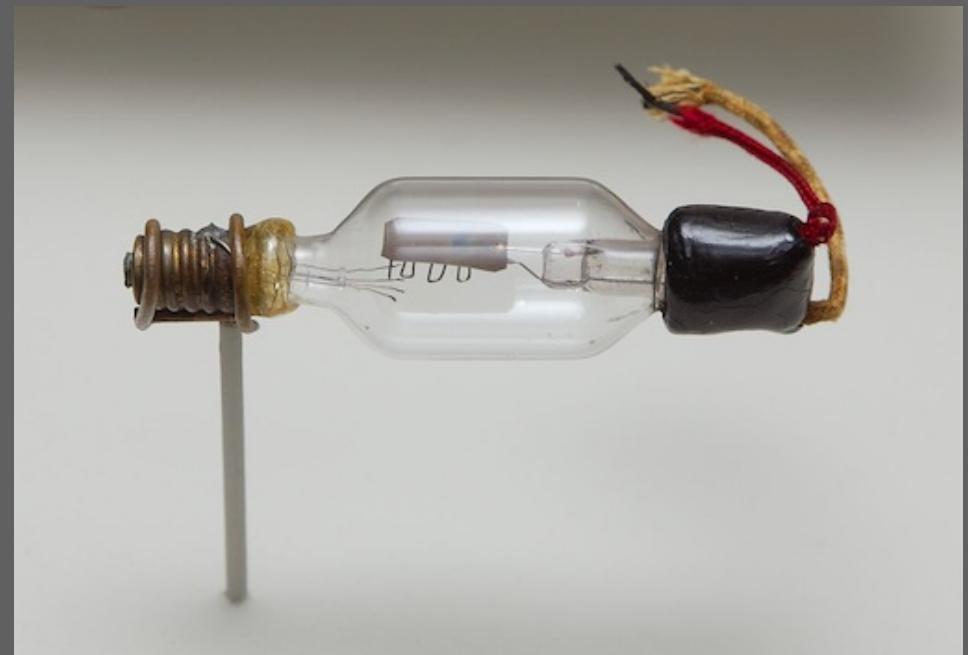
Le valvole termoioniche o “tubi a vuoto”

- Ma per la storia dei calcolatori **elettronici** è fondamentale anche l’invenzione del **triodo**, avvenuta nel 1906 grazie all’americano **Lee De Forest** (nel 1904 era stato inventato il diodo).
- Il triodo è il primo esemplare di **componente elettronico attivo**, ossia in grado di **amplificare** un segnale elettrico (mentre il diodo è passivo).
- Ha aspetto (e dimensioni) simili a quelle di una lampadina ad incandescenza, all’interno della quale è quindi stato fatto il vuoto, e in cui un filamento (il catodo) viene riscaldato mediante corrente elettrica



Le valvole termoioniche o “tubi a vuoto”

- Il riscaldamento del catodo produce una emissione di elettroni (quindi una corrente elettrica) che fluisce dal catodo verso l'anodo, attraversando la griglia.
- Tuttavia, se anche la griglia è sotto tensione, il flusso di elettroni che riesce ad attraversarla dipende dal livello di tensione applicato alla griglia stessa (o meglio, alla differenza di tensione tra griglia e catodo)
- Dunque applicando una (piccola) tensione *variabile* alla griglia, la quantità di corrente che fluisce tra catodo e anodo seguirà l'andamento del segnale applicato alla griglia, ma amplificato.



Le valvole termoioniche o “tubi a vuoto”

- Nel corso degli anni, sono stati progettati diversi tipi di valvole termoioniche, oltre ai diodi e ai triodi: tetrodi, pentodi, e addirittura esodi, eptodi, ottodi, più comunemente detti valvole multigriglia, usate soprattutto per le applicazioni radio.
- È stata migliorata la tecnologia costruttiva, in modo ad esempio da renderle più affidabili e da consumare meno energia elettrica.
- Hanno dominato l'intero panorama dell'era elettronica per molti anni, e hanno avuto gli impieghi più disparati, in particolare nelle radio e nelle televisioni.
- Ormai quasi completamente sostituite dalla tecnologia che sfrutta i semiconduttori, le valvole termoioniche rimangono ancora in produzione e in uso nel campo dell'altra fedeltà, poiché permettono di costruire amplificatori musicali di altissima qualità sonora. ⁵³

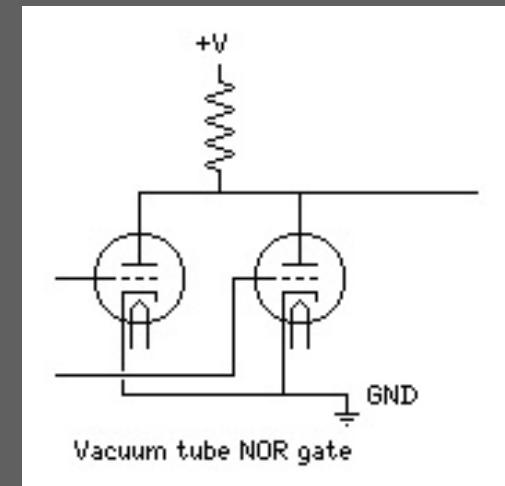
Le valvole termoioniche o “tubi a vuoto”

- Un esempio di amplificatore moderno a valvole termoioniche (che vengono usate anche come elemento estetico).
- Ogni esemplare viene costruito a mano, e il prezzo supera facilmente i 10.000 euro.



Le valvole termoioniche o “tubi a vuoto”

- Scelti opportunamente, due soli diversi valori di tensione alla griglia di un triodo permettono di usarlo come un interruttore: la corrente fluisce o meno tra catodo e anodo a seconda della tensione applicata alla griglia.
- I vantaggi rispetto ad un relè sono evidenti, perché non ci sono parti meccaniche in gioco (quindi maggior velocità di commutazione e minor usura) e il consumo di corrente per azionare l'interruttore è inferiore a quello richiesto ad una elettrocalamita.
- Una porta NOR realizzata con due triodi.
Gli input della porta (sulla sinistra) sono costituiti dalle griglie dei due triodi, mentre l'output (sulla destra) è costituito dai due anodi collegati in parallelo.



I primi computer digitali: il Model K

- Tuttavia, nonostante le valvole termoioniche fossero state inventate da tempo, i primi esperimenti di calcolo digitale avvennero utilizzando i relè (in primo luogo perché più economici).
- Nel 1937 l'americano **George Sibitz**, impiegato ai laboratori Bell, intuisce che la logica booleana poteva essere usata per costruire circuiti digitali in grado di compiere operazioni aritmetiche.
- Utilizzando dei relè costruisce nella cucina di casa il **Model K** (K per *Kitchen*), un addizionatore di numeri a due cifre binarie.
- Sulla base di questa esperienza, tra il 1937 e il 1939 progetta e costruisce il **Complex Number Calculator**, composto da 450 relè e in grado di eseguire le quattro operazioni sui numeri complessi (sotto la pressione della guerra, seguiranno poi altri modelli, più potenti e parzialmente programmabili, sempre basati sull'uso di relè)

I primi computer digitali

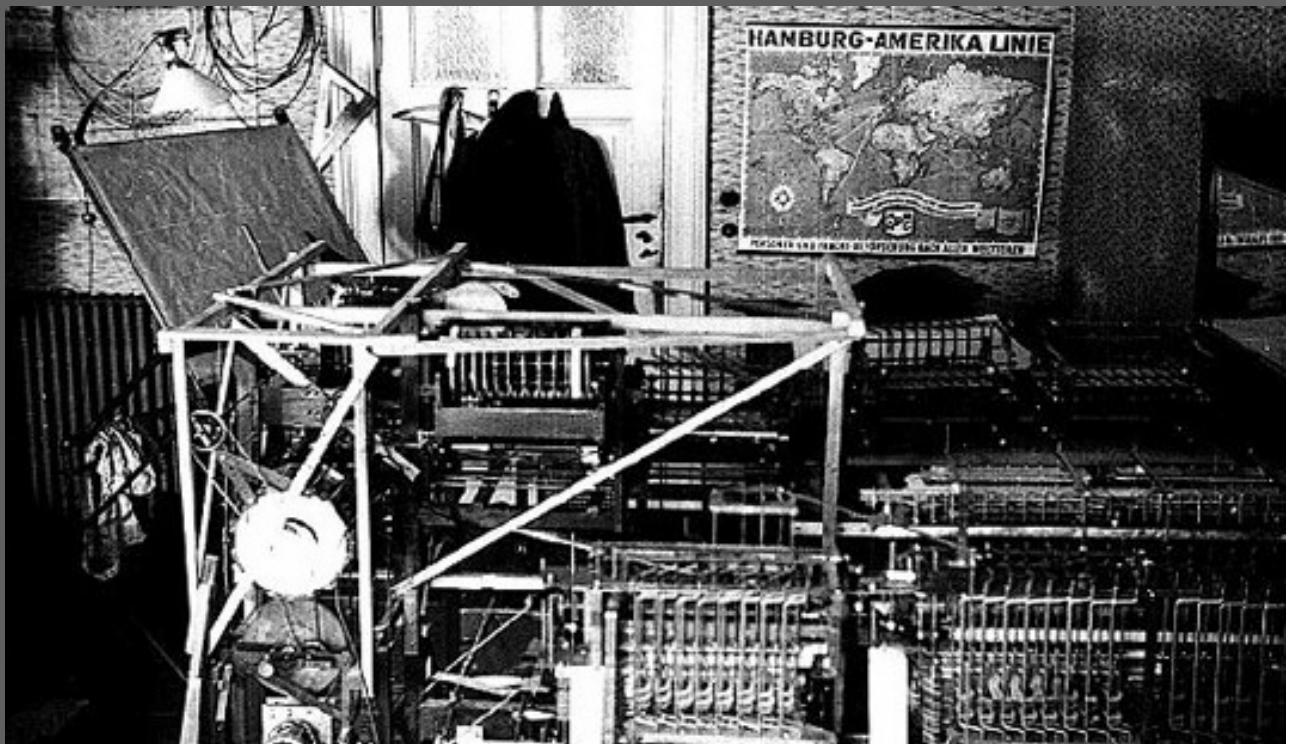
- In effetti, nella seconda metà degli anni 30, quando i computer analogici erano da tempo in uso, cominciava a prendere piede l'idea di usare la logica booleana per costruire calcolatori digitali.
- Più o meno nello stesso periodo infatti l'americano **Claude Shannon** (il padre della **teoria dell'informazione**) aveva studiato la corrispondenza tra la logica booleana e i circuiti digitali a relè, discutendo la sua tesi di laurea su questo argomento nel 1938.
- Shannon andò poi a lavorare ai laboratori Bell, e lui e Stibitz erano al corrente dei rispettivi lavori, anche se non pare che Shannon abbia preso parte al progetto del Complex Number Calculator
- Stibitz invece non sapeva che nello stesso momento, a Berlino, qualcuno stava compiendo degli esperimenti simili ai suoi.

I primi computer digitali: lo Z1

- **Konrad Zuse**, ingegnere civile, cercava un modo per eseguire in modo automatico i laboriosi calcoli necessari nella progettazione di edifici e strade, e pensò di realizzare una macchina che fosse sufficientemente flessibile e programmabile.
- Anch'egli ebbe l'idea di usare la numerazione binaria, che si prestava facilmente ad essere realizzata (rappresentazione dei numeri e operazioni fra di essi) attraverso semplici dispositivi meccanici che potevano assumere due diverse posizioni (l'uso di interruttori a relè verrà adottato poco più avanti).
- Progetta e realizza la sua prima macchina dal 1936 al 1938 nel salotto dei genitori e la chiama “V1” (da *versuchsmodell* = modello sperimentale). Dopo la seconda guerra mondiale la V dei vari modelli verrà modificata in Z, e quindi la V1 diviene la **Z1**.

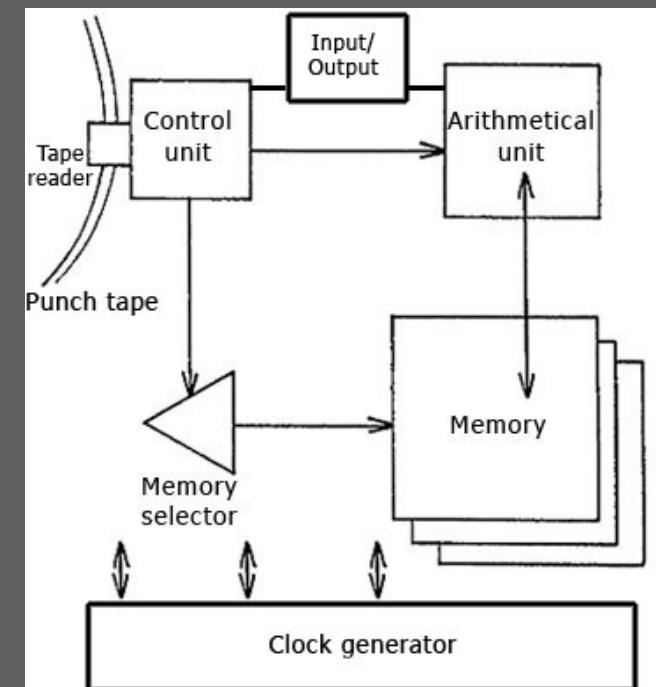
I primi computer digitali: lo Z1

- La Z1 era completamente meccanica, ma era “mossa” da un motore elettrico da 1 kw che faceva avanzare la computazione alla velocità di un passo al secondo (oggi diremmo un ciclo di clock lungo un secondo, o una frequenza del clock di 1 Hz)
- Era composta da circa 20.000 parti tutte costruite a mano mediante un seghetto da Zuse e i suoi amici, e pesava circa 1000 kg.
- Una manovella permetteva di farla funzionare anche a mano.
- Eseguiva una moltiplicazione in 5 secondi



I primi computer digitali: lo Z1

- La Z1 è il primo esempio di macchina con una struttura interna concettualmente simile a quella dei moderni computer:
- **Memory:** 64 word da 22 bit indirizzate dal **Memory Selector**
- **Arithmetical Unit:** con 2 registri che potevano essere sommati o sottratti (tutte le operazioni erano ricondotte a somme/sottrazioni)
- **Lettore di nastro perforato:** che leggeva le istruzioni da eseguire
- **Control Unit:** che gestiva l'esecuzione delle istruzioni
- **Memory Selector:** per indirizzare la Memory
- Una tastiera per inserire **L'input** in forma decimale (automaticamente convertita in binario)
- Un dispositivo di **output** per convertire in decimale il contenuto di R1



I primi computer digitali: lo Z1

- Il nastro perforato conteneva la sequenza di istruzioni da eseguire, codificate su 8 bit, e questo era l'**instruction set**:
- **Lu**: to call the input device for decimal numbers
- **Ld**: to call the output device for decimal numbers
- **Pr1 z**: read the contents of the memory cell z into Registers $R1$
- **Pr2 z**: read the contents of the memory cell z into Registers $R2$
- **Ps z**: write the contents of Register $R1$ to the memory cell z
- **Ls1**: add the two numbers in the Registers $R1$ and $R2$ (*dest.* = $R1$)
- **Ls2**: subtract the two numbers in the Registers $R1$ and $R2$
- **Lm**: multiply the two numbers in the Registers $R1$ and $R2$
- **Li**: divide the two numbers in the Registers $R1$ and $R2$

I primi computer digitali: lo Z1

- In effetti, già nel 1936 un amico di Zuse, **Helmut Schreyer**, gli aveva suggerito di costruire lo Z1 usando i tubi a vuoto, ma a Zuse l'idea sembrò strana, dato che al tempo erano usati per gli apparecchi radio.
- Schreyer andò avanti con la sua idea, e nella sua tesi di dottorato mostrò come creare diversi circuiti logici usando le valvole.
- A partire dal 1939, persuaso dal suo amico, Zuse e Schreyer cercano di interessare altri scienziati berlinesi e le autorità a finanziare la costruzione di un computer per l'aviazione militare costruito usando valvole termoioniche.
- Tuttavia, il loro progetto prevedeva circa 2000 valvole (un numero enorme per i dispositivi elettronici del tempo), e un tempo di costruzione stimato in due anni, sicché la risposta delle autorità fu: *vinceremo la guerra molto prima che il suo computer sia pronto, per cui, a che pro?*

I primi computer digitali: lo Z1

- Nel 1937 Zuse, alla ricerca di fondi per completare lo Z1, aveva contattato Kurt Pannke, un costruttore di calcolatori meccanici, il quale però, a proposito del lavoro su cui era impegnato Zuse, fece la seguente osservazione:

...at the field of computing machines virtually everything, until the last possible approaches and sophisticated devices, has already been invented. There is hardly anything left to invent.
- Tuttavia, impressionato dal lavoro di Zuse sullo Z1 concede un prestito, e questo permetterà a Zuse, nel 1939, di sviluppare lo Z2, una versione dello Z1 in cui l'unità di controllo utilizzava 200 relè.
- Seguirà nel 1941 lo Z3, completamente realizzato usando relè (circa 2400), e a partire dal 1942 il progetto dello Z4, completato tuttavia solo nel 1950 dopo svariate peripezie dovute alle vicende della guerra.

I primi computer digitali: il Mark I

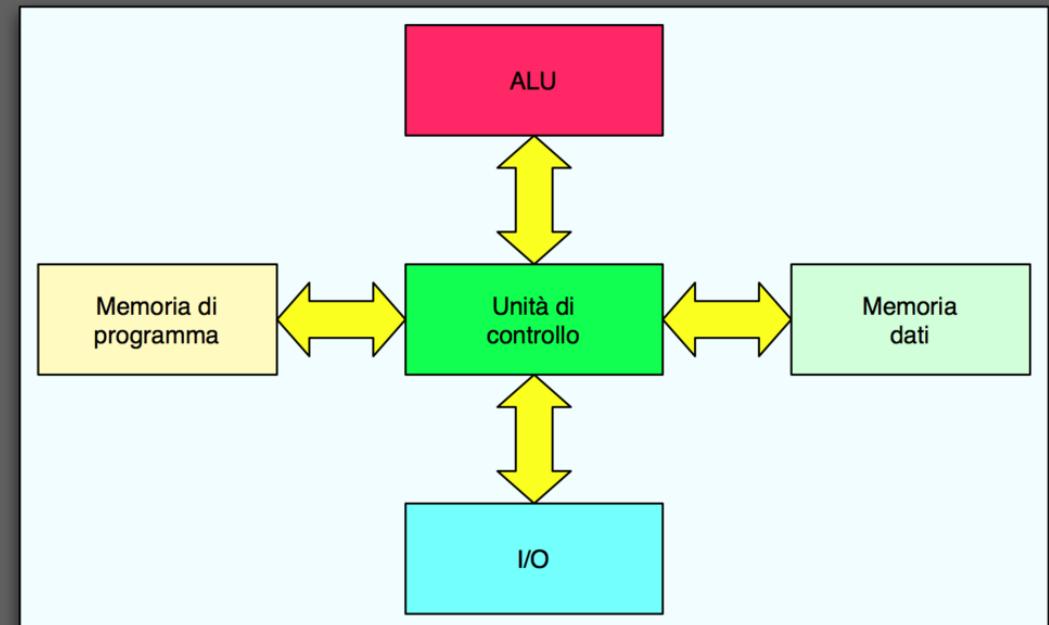
- Nel 1937 il fisico americano **Howard Aiken** dell'università di Harvard, che aveva studiato il lavoro di Babbage pensa di costruire una macchina simile all'analytical Engine usando però i relè.
- Contatta l'IBM e riceve un finanziamento e supporti tecnici che portarono alla costruzione **dell'Automatic Sequence Controlled Calculator**, o **ASSC**, più avanti rinominato **Mark I**.
- (In realtà aveva prima contattato altre compagnie, tra cui la più grande compagnia di costruzione di calcolatrici meccaniche del tempo, la Monroe Calculator Company, ma ricevendo sempre risposte negative)
- Il calcolatore fu completato solo nel 1943, ad un costo di 250 mila dollari del tempo, e benché reso velocemente obsoleto dai calcolatori a componenti elettronici, rimase in funzione fino al 1959

I primi computer digitali: il Mark I

- Il Mark I era costruito usando relè, alberi di rotazione e frizioni. I meccanismi erano messi in movimento da un motore da 4 KW
- Era composto da circa 765 mila pezzi, pesava più di 4 tonnellate ed era lungo circa 15 metri, alto 2,4 metri e profondo mezzo metro.
- Di fatto era composto da 78 calcolatrici elettromeccaniche sincronizzate fra loro da un albero di rotazione lungo 15 metri.
- Eseguiva programmi leggendo ed eseguendo una per una le istruzioni scritte su schede o nastro perforato, e poteva memorizzare 72 numeri di 23 cifre decimali ciascuno.
- In effetti era piuttosto lento, dato che poteva eseguire 3 addizioni o sottrazioni al secondo, ma eseguiva una moltiplicazione in 6 secondi, una divisione in 15 secondi, e ci voleva più di un minuto per calcolare un logaritmo.

I primi computer digitali: il Mark I

- Il Mark I è ricordato soprattutto perché è da esso che discende il concetto di **architettura Harvard**, con la quale si intende una architettura di computer che adotta una netta separazione tra la memoria che contiene i dati e la memoria che contiene le istruzioni.
- I computer moderni adottano la cosiddetta architettura **Von Neumann** ma l'architettura Harvard è usata ad esempio nei microcontrollori per applicazioni industriali, in cui il programma è memorizzato su una ROM e i dati su una RAM, e vi una separazione netta fra il flusso dei dati e delle istruzioni
- Però a voler essere pignoli, nel datapath di un processore moderno...



I primi computer digitali: il Mark I

- Il Mark I fu causa di un incidente diplomatico fra IBM e l'università di Harvard, che alla presentazione della macchina glissò sul contributo fondamentale dato dalla IBM alla progettazione e costruzione del calcolatore.
- Ciò nonostante, fu grazie all'esperienza sviluppata col Mark I che la IBM decise di dedicarsi alla costruzione di calcolatori elettronici.



Una nota di colore: il primo bug!

- Al Mark I seguirono altri modelli, e in particolare il **Mark II**, completato nel 1947 e costruito usando relè elettromagnetici, molto più veloci di quelli elettromeccanici impiegati nel Mark I
- Era in grado di eseguire una somma in 0,125 secondi, e una moltiplicazione in 0,75 secondi.
- Nel 1945, mentre lavorava su un prototipo del Mark II, **Grace Murray Hopper**, programmatrice del Mark I, scopre il primo bug (insetto, cimice) in un computer: Una tarma che aveva causato il malfunzionamento di uno dei relè.
- È da questo episodio che nasce l'espressione **“computer bug”** usata universalmente nel gergo informatico (**“baco”** in italiano)

I primi computer digitali: l'ABC

- L'**Atanasoff-Berry Computer** (da cui appunto l'acronimo **ABC**) è stato il primo computer realizzato completamente con componenti elettronici (come tra l'altro sancito da un tribunale americano nel 1973 in una disputa con il più famoso ENIAC).
- Era costituito di sole valvole termoioniche (più di 300) e altri componenti elettrici, mentre era completamente assente qualsiasi componente meccanico o elettromeccanico.
- Alimentato da tensione alternata a 60 Hz (che costituiva anche il clock della macchina), pesava circa 320 Kg, impiegava 1,6 Km di cavi e aveva la dimensione di una scrivania.
- Fu progettato e realizzato tra il 1937 e il 1941 da **John Atanasoff** e **Clifford Berry** all'università dell'Iowa, per la risoluzione di equazioni lineari.

I primi computer digitali: l'ABC

- Non era quindi un vero e proprio computer programmabile (più formalmente non era **Turing completo**, un concetto comunque non ancora chiaro all'epoca), e la programmazione era parzialmente manuale, richiedendo l'intervento di operatori che agivano su switch e collegamenti a seconda dell'equazione da risolvere.
- Usava una rappresentazione completamente binaria dei dati, eseguendo fino 30 somme/sottrazioni al secondo su numeri di 50 bit.
- Di particolare interesse era la memoria dati. In effetti, il **flip-flop** elettronico era stato inventato già nel 1918 **William Eccles** e **Frank Jordan** usando tubi a vuoto.
- Atanasoff e Berry usaroni invece per memorizzare i bit di dati 1600 condensatori, la cui carica veniva periodicamente rinfrescata, proprio come nelle moderne **DRAM!**

La seconda guerra mondiale: le Bombe

- Lo scoppio della seconda guerra mondiale ostacola enormemente il lavoro di ricercatori come Atanasoff e Zuse.
- Ha tuttavia anche l'effetto di spingere ai primi passi della ricerca nel campo dell'elaborazione elettromeccanica ed elettronica la Gran Bretagna. Lo scopo era di decifrare i messaggi criptati usati dall'esercito tedesco per le comunicazioni.
- A partire dal 1925 l'esercito tedesco comincia ad usare la famigerata macchina **Enigma** per criptare i messaggi. Ogni messaggio poteva essere codificato in più di 100.000 messaggi cifrati diversi, per cui la codifica era ritenuta sicura.

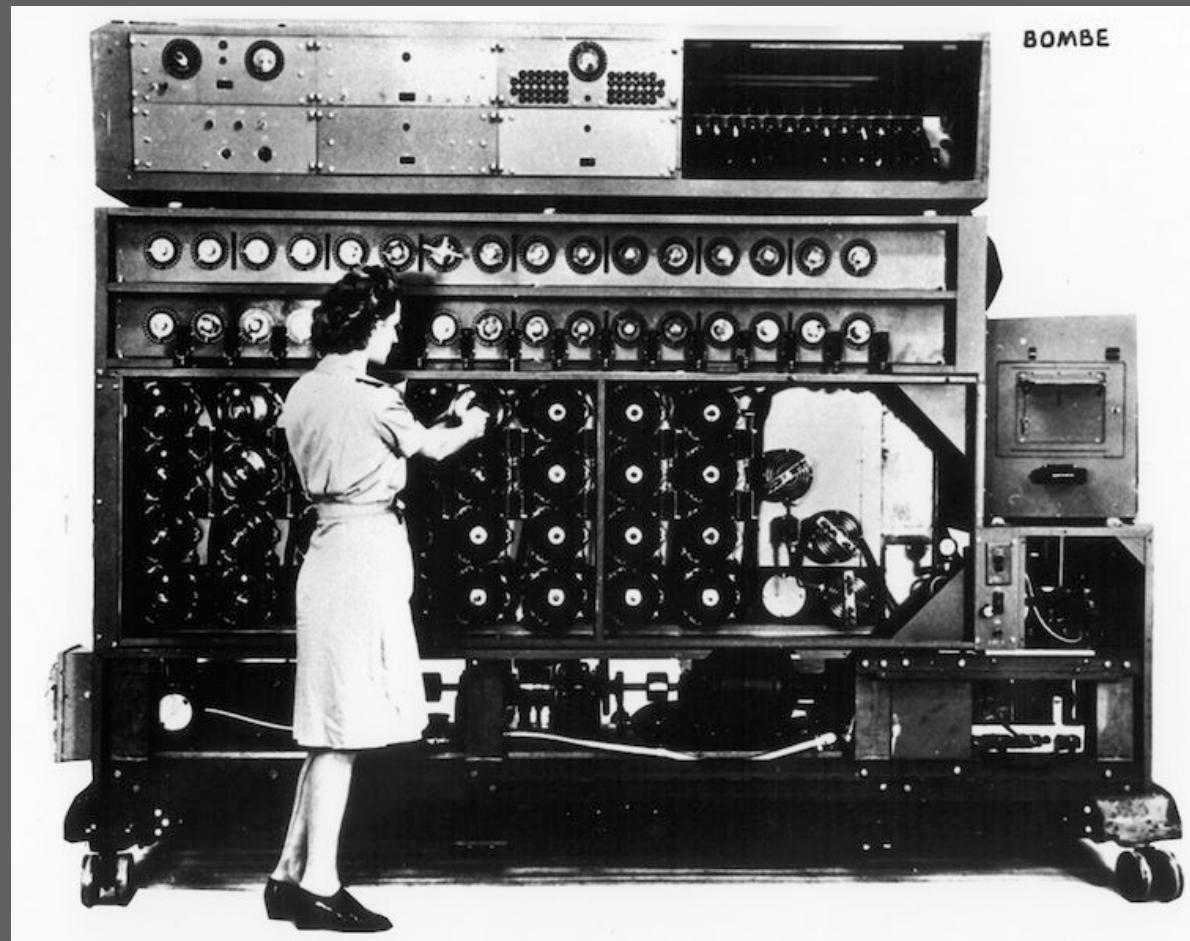


La seconda guerra mondiale: le Bombe

- Non di meno, l'intelligence polacca utilizzando una macchina programmabile elettromeccanica chiamata **Bomba** riesce nel 1932 a decrittare i messaggi codificati con Enigma.
- Come conseguenza, e in occasione della guerra, Enigma viene modificata più volte, portando fino ad un massimo di 18×10^{19} il numero di codifiche diverse possibili.
- Nel 1939 l'intelligence polacca fa avere a inglesi e francesi tutto quanto sa su Enigma e gli schemi e i progetti della “bomba” e due matematici inglesi, **Gordon Welchman** e **Alan Turing** riescono a sviluppare una versione più sofisticata della bomba in grado di decrittare i messaggi della versione più complessa di Enigma.

La seconda guerra mondiale: le Bombe

- La bomba simulava sostanzialmente il funzionamento di più macchine enigma in parallelo, esplorando così diverse possibili decodifiche dei messaggi criptati.
- La versione migliorata da Welchman e Turing cominciò ad operare circa a metà del 1940, permettendo la decodifica di 178 messaggi criptati dell'esercito nazista.



La seconda guerra mondiale: i Colossus

- A partire dal 1941 tuttavia, gli inglesi intercettano messaggi che non riescono a decifrare con le “bombe”; sono infatti stati codificati dai nazisti in modo diverso, la **cifratura di Lorenz**.
- Gli inglesi allora assegnano all’ingegnere **Tommy Flower**, i compito di costruire una macchina in grado di decifrare i messaggi criptati.
- Sfruttando anche i lavori di Turing sull’uso delle probabilità nel campo della criptoanalisi, Flower progetta e costruisce un macchina completamente elettronica (lettore di nastro perforato a parte), il **Colossus Mark 1** che dall’inizio del 1944 riuscirà a decodificare i messaggi in codice dei nazisti.

La seconda guerra mondiale: i Colossus

- Benché parzialmente programmabile, il Colossus non era ancora un computer nel senso moderno del termine, in quanto non era Turing completo. Tuttavia, all'università di San francisco B. Wells ha dimostrato nel 2008 che una macchina di Turing universale poteva essere simulata usando 10 Colossus in parallelo.
- Il successore del Mark 1, il **Mark 2**, costruito a partire dal 1944 in 10 esemplari, era 5 volte più veloce del Mark 1: era composto da 2400 tubi a vuoto, 800 relè, e poteva elaborare circa 5.000 caratteri del testo da decifrare al secondo.

Verso la fine della guerra e i computer moderni: la storia dell'ENIAC

- Con l'ingresso in guerra, l'esercito americano aveva un urgente bisogno di calcolare nuove tavole balistiche necessarie per rendere precisi i colpi dei cannoni a lunga gittata usati nei territori europei e africani.
- Il punto esatto in cui un proiettile sarebbe caduto dipendeva da molte variabili: la velocità del vento e la sua direzione, l'umidità e la temperatura dell'aria, l'altezza sul livello del mare e la conformazione del terreno, e così via.
- Ogni modello di cannone e proiettile aveva bisogno di nuove tavole balistiche, in cui i calcoli per le diverse condizioni venivano fatti combinando formule matematiche con i risultati di test di fuoco.⁷⁶

la storia dell'ENIAC

- L'esercito USA impiegava uno staff di 176 persone (chiamate "computers") le quali usavano calcolatrici meccaniche da tavolo a manovella.
- Avevano a disposizione anche 2 computer analogici in grado di risolvere equazioni differenziali (ma che erano lenti e spesso fuori uso).
- Tuttavia, era necessario molto tempo per compilare la compilazione di una singola tavola balistica, che conteneva 3000 traiettorie ognuna delle quali richiedeva circa 750 calcoli per essere computata!
- Senza queste tavole, le armi inviate in Europa e Africa sarebbero risultate inutilizzabili.



la storia dell'ENIAC

- Una parte dei “computers” lavoravano alla **Moore School of Electrical Engineering** della Università della Pennsylvania, sotto la direzione di un giovane professore di matematica di 29 anni, il tenente **Herman Goldstine**.
- Addirittura, Goldstine aveva mandato in giro per la Pennsylvania sua moglie, essa stessa professoressa di matematica, nel tentativo di reclutare brave studentesse di matematica che potessero aiutare a velocizzare i calcoli.
- Un giorno, uno studente di Goldstine gli segnala il lavoro di un altro professore che era appena stato assunto nella stessa scuola, **John Mauchly**, il quale aveva avuto l’idea di sostituire tutti i “computers” umani con un calcolatore elettronico.

la storia dell'ENIAC

- Già nel 1938 Mauchly aveva incominciato a sperimentare la costruzione di piccoli circuiti elettronici digitali usando tubi a vuoto.
- In particolare, era riuscito a costruire dei flip-flop, in grado di memorizzare singoli bit di informazione (una tecnica usata ancora oggi nei moderni processori)
- Nel 1942 Mauchly, già alla Moore School, scrive un memorandum dal titolo: *The Use of High-Speed Vacuum Tube Devices for Calculation* in cui osserva tra l'altro:

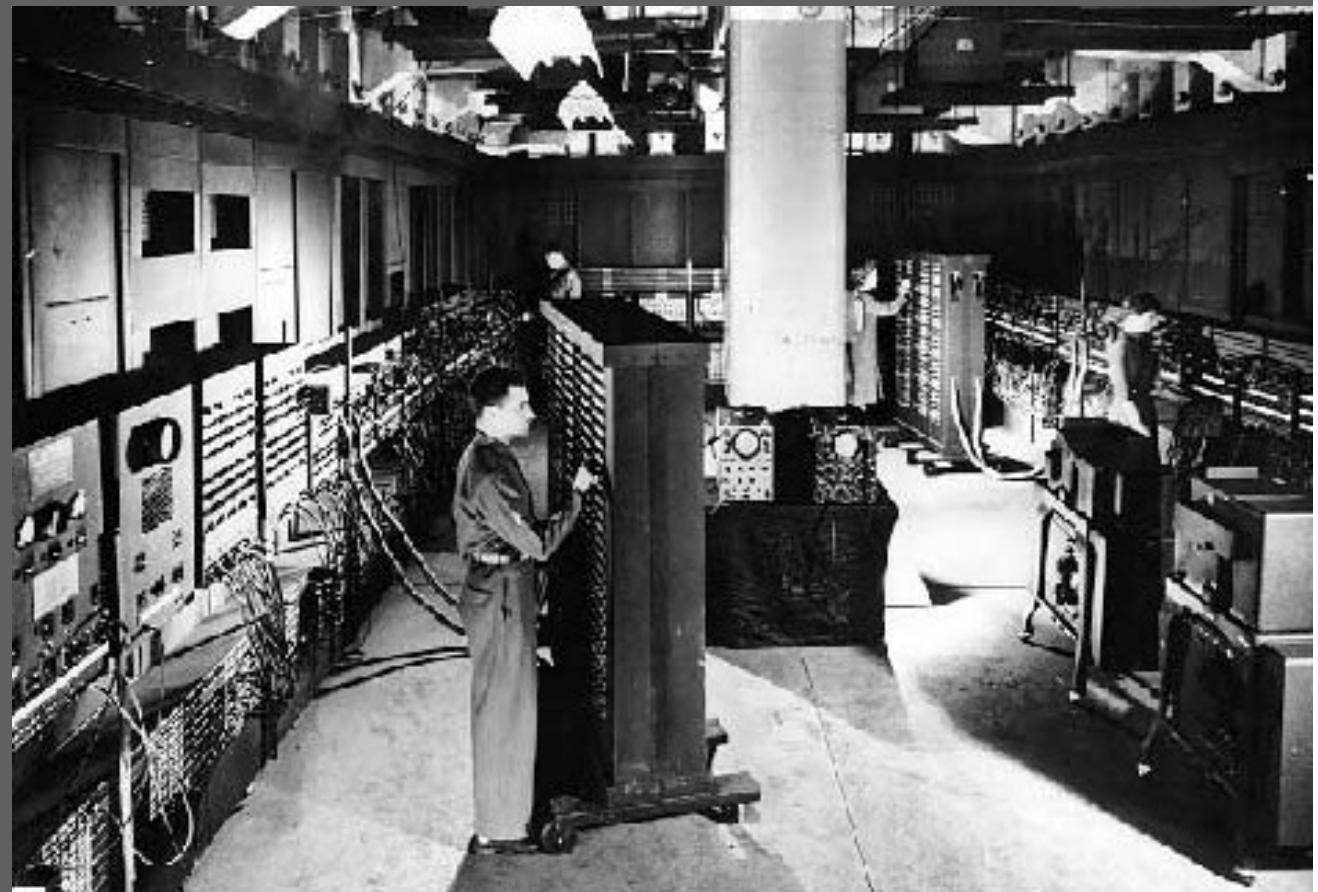
A great gain in the speed of the calculation can be obtained if the devices that are used employ electronic means for the performance of the calculation, because the speed of such devices can be made very much higher than that of any mechanical device

la storia dell'ENIAC

- Goldstine contatta Mauchly e presenta lui e **John Eckert** (appena laureatosi alla Moore School) all'ufficiale direttore del Laboratorio di Ricerca Balistica.
- Viene così stipulato un contratto tra l'esercito e l'università della Pennsylvania per un periodo iniziale di sei mesi per lo sviluppo di un Electronic Numerical Integrator, nome che cambierà poi in **Electronic Numerical Integrator and Computer (ENIAC)**.
- I lavori di costruzione veri e propri incominciano nel giugno del 1943 sotto la direzione operativa di Eckert e la supervisione di Goldstine.
- I lavori terminarono solo nell'autunno del 1945, troppo tardi per la guerra, e l'ENIAC fu presentato al pubblico solo nel febbraio del 1946

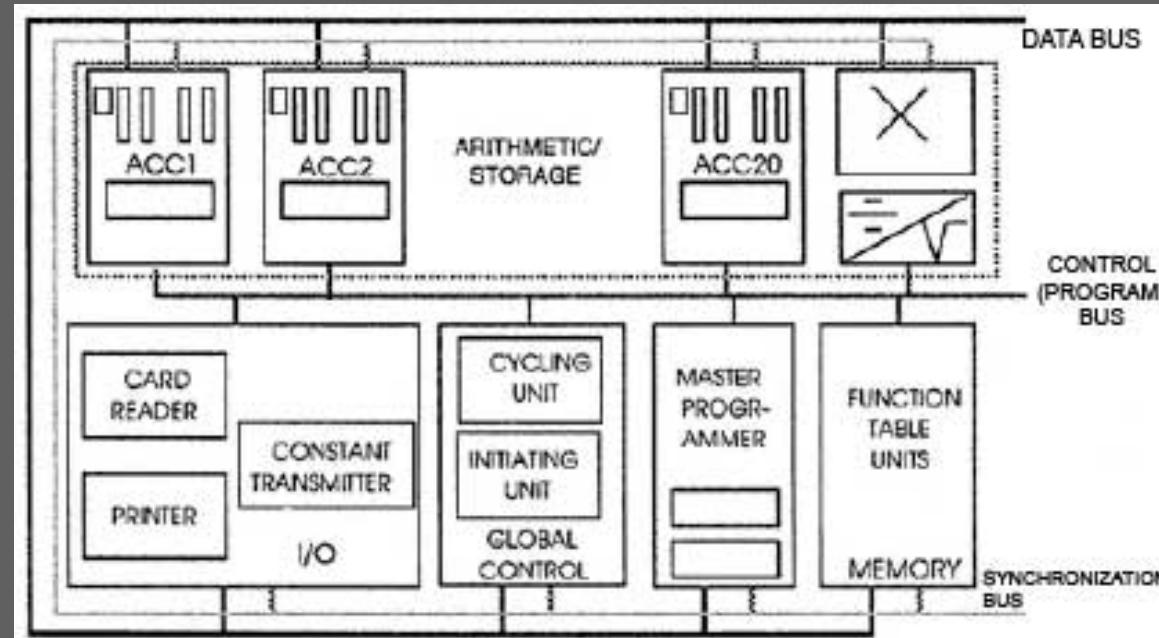
la storia dell'ENIAC

- Ne risulta un mostro da 27 tonnellate, che occupa una stanza di 10×15 metri, e che ha richiesto 200.000 ore-uomo di lavoro, e 587.000 \$ (circa 6 milioni di dollari attuali).
- È formato da 17.468 tubi a vuoto, 1500 relè, 10.000 condensatori, 70.000 resistenze, 5.000.000 di punti di saldatura fatti a mano e consuma 174 KW.
- I guasti erano molto frequenti, poiché i tubi erano delicati, e potevano rompersene molti ogni giorno...
- Nella figura vediamo Goldstine che opera sul pannello di controllo dell'ENIAC



la storia dell'ENIAC

- L'ENIAC, aveva una architettura molto diversa da quella dei moderni computer, e **non usava ancora un programma memorizzato**.
- in sostanza era un insieme di venti macchine addizionatrici (gli “accumulatori”) e altre unità aritmetiche controllate da un enorme rete di cavi elettrici e switch **riconfigurabili manualmente**
- L'input era costituito da un lettore di schede perforate (ma si veda più avanti), e l'output era costituito da un perforatore di schede.



la storia dell'ENIAC

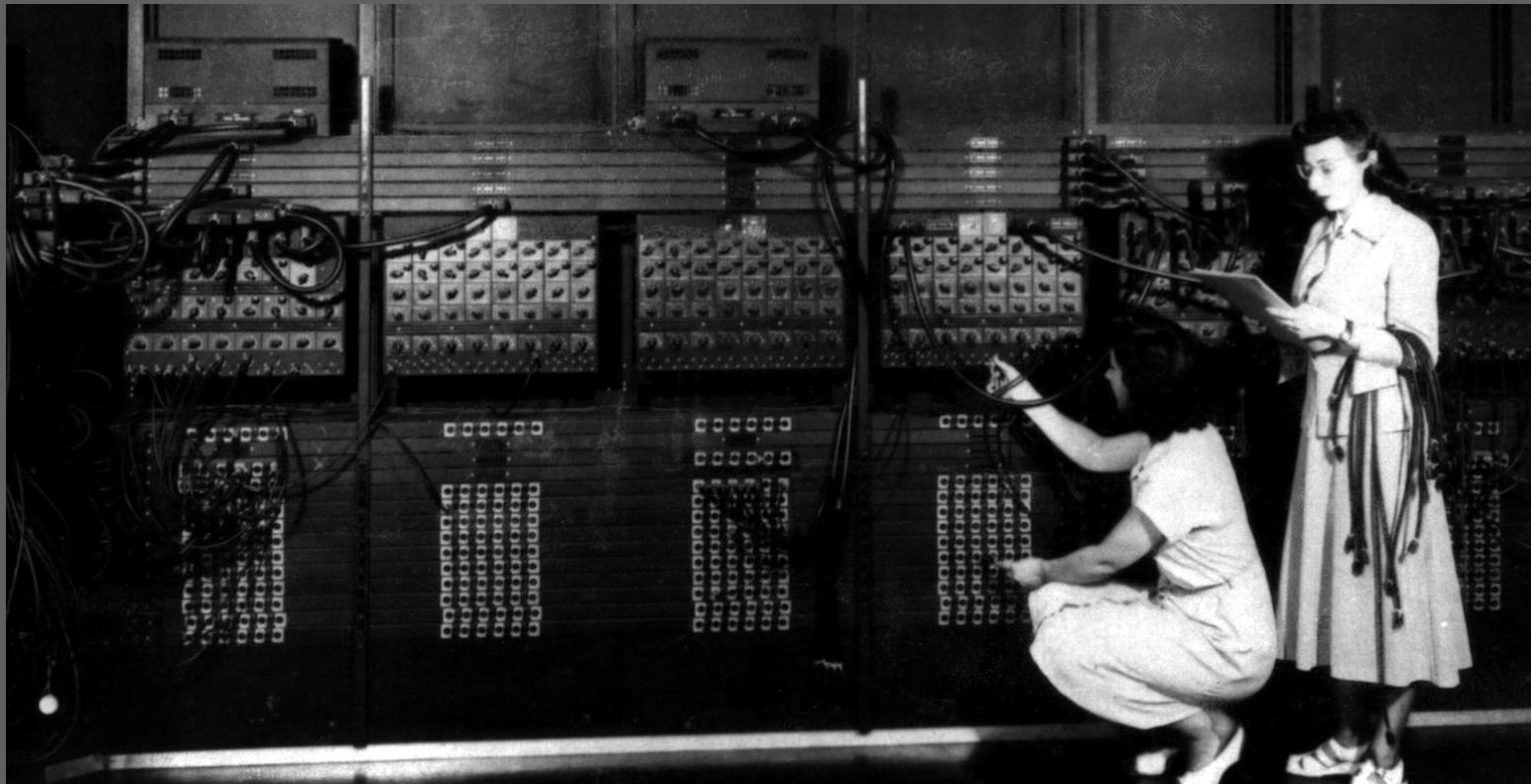
- Ogni accumulatore era costituito da 36 tubi a vuoto, e aveva l'aspetto di un pannello con una dimensione di circa un metro quadrato.
- era in grado di memorizzare un numero in base 10 formato da 10 cifre, sommare/sottrarre il proprio contenuto con quello di un altro accumulatore, e memorizzare il risultato.
- Se gli accumulatori venivano usati tutti insieme in parallelo era possibile eseguire fino a 5000 somme/sottrazioni al secondo.
- 4 accumulatori potevano essere collegati fra loro in modo opportuno (notate, la cosa veniva fatta a mano...) e, controllati da una speciale unità di moltiplicazione, potevano eseguire fino a 385 moltiplicazioni al secondo.
- Analogamente, 5 accumulatori collegati opportunamente permettevano di eseguire 40 divisioni o 3 radici quadrate al secondo.

la storia dell'ENIAC

- Può essere interessante notare come l'ENIAC non usasse una rappresentazione binaria dell'informazione, almeno nel senso in cui lo intendiamo noi.
- Ogni cifra di un numero in base 10 veniva rappresentata usando 10 flip flop (costruiti con tubi a vuoto) in fila e numerati da 0 a 9.
- Se ad esempio si doveva rappresentare la cifra 4, veniva messo a 1 il flip flop numero 4, e tutti gli altri rimanevano a 0.

la storia dell'ENIAC

- L'ENIAC aveva un ciclo di clock di 200 microsecondi, nel quale era appunto in grado di eseguire una somma/sottrazione o di memorizzare un numero in uno specifico accumulatore.
- L'ENIAC veniva programmato riconfigurando opportunamente a mano gli switch e i collegamenti elettrici, e poteva eseguire salti condizionati e cicli.



la storia dell'ENIAC

- **Prendere un problema da risolvere e mapparlo nell'ENIAC era un compito molto complicato, che richiedeva settimane!**
1. Il problema veniva formalizzato in un insieme di equazioni.
 2. Le equazioni venivano scomposte in una sequenza di operazioni aritmetiche che l'ENIAC era in grado di eseguire.
 3. Alcuni dei dati numerici di input venivano forniti all'ENIAC mediante un lettore di schede perforate (si veda più avanti).
 4. “Programmare” l'ENIAC significava decidere per ogni accumulatore quale operazione aritmetica doveva eseguire, da quale altro accumulatore doveva ricevere l'input, e a quale accumulatore trasmettere l'output.
 5. La programmazione di ogni accumulatore avveniva riconfigurando manualmente i collegamenti elettrici e gli switch di cui ogni accumulatore era dotato.

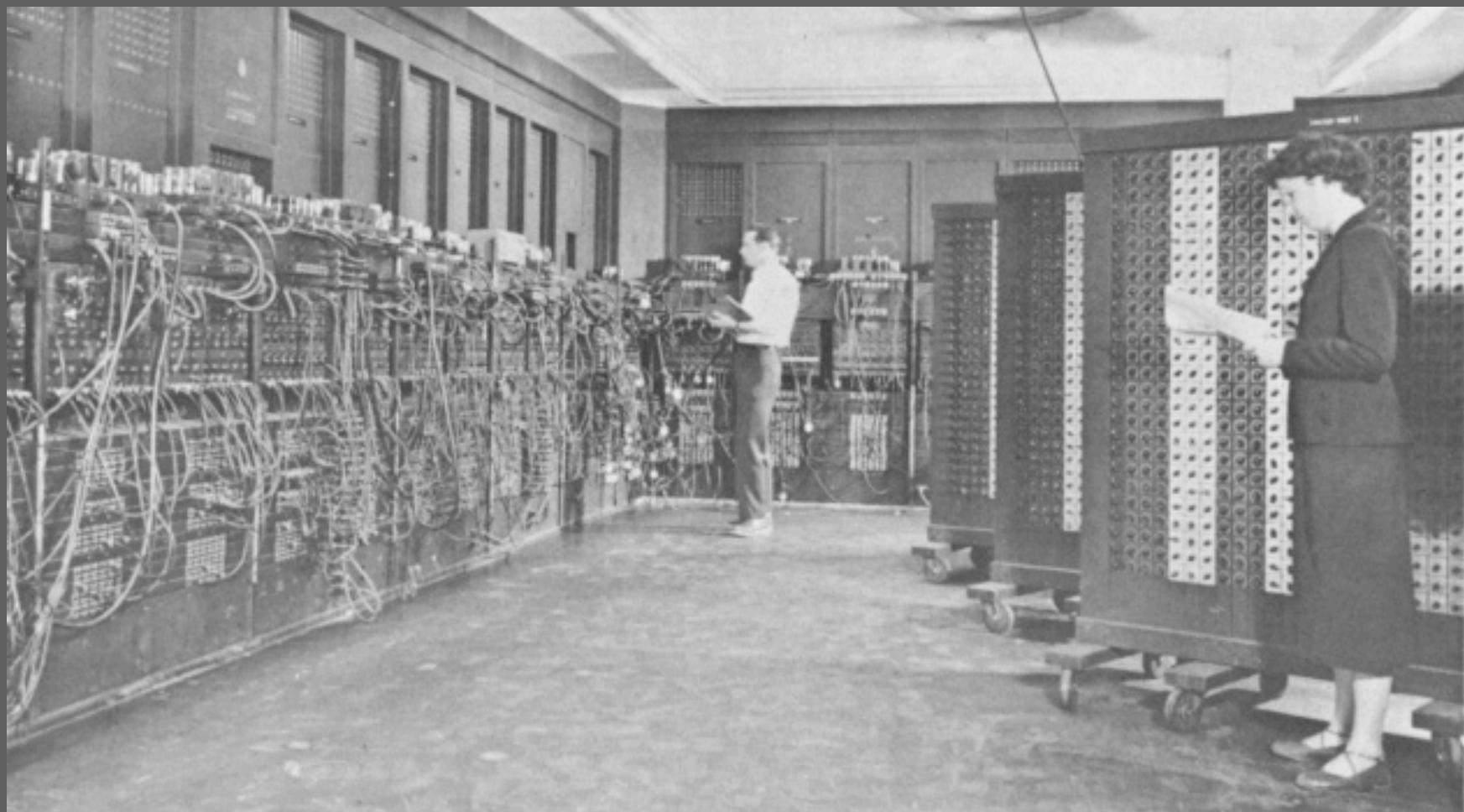
la storia dell'ENIAC

6. Il programma da eseguire vero e proprio consisteva nell'esecuzione in sequenza delle operazioni programmate su ogni accumulatore.
7. L'ordine di esecuzione veniva a sua volta “programmato” operando, sempre manualmente, su una serie di switch e collegamenti elettrici del master programmer (inclusi eventuali loop)
8. La capacità di memorizzazione dei dati di input di un problema era chiaramente limitata ai 20 accumulatori, che dovevano memorizzare anche i valori intermedi delle variabili coinvolte nel calcolo.

per questa ragione, ulteriori dati numerici di input venivano resi disponibili, anziché con le schede perforate, mediante 3 (o 4) “function tables”, sostanzialmente dei pannelli mobili ciascuno formato da 1200 switch a 10 posizioni, ognuna in grado di memorizzare 104 numeri a 10 cifre con segno.

la storia dell'ENIAC

- In questa figura vediamo sulla sinistra gli accumulatori programmati mediante cavi elettrici, e sulla destra 4 function tables mobili che, una volta configurate, potevano essere collegate agli accumulatori per fornire i dati numerici addizionali di un problema.



la storia dell'ENIAC

- Poiché la “programmazione” dell’ENIAC per risolvere un certo problema richiedeva la configurazione di migliaia di switch e cavi elettrici, una volta programmato si cercava di farlo lavorare su quel problema con tutti i dati possibili, prima di passare ad un problema diverso e quindi “riprogrammarlo”
- Quasi tutta la programmazione manuale dell’ENIAC veniva fatta da sei donne: **Kay McNulty**, **Betty Jennings**, **Betty Snyder**, **Marlyn Wescoff**, **Fran Bilas** e **Ruth Lichterman**, che nel 1997 sono state inserite nella **Women in Technology Hall of Fame** dopo essere state ignorate per circa 40 anni... (vediamo alcune di loro nelle immagini dei lucidi precedenti)

la storia dell'ENIAC

- Sebbene la programmazione dell'ENIAC per risolvere un problema potesse richiedere settimane, una volta programmato poteva fornire la soluzione **per una qualsiasi configurazione dei dati di input** in pochi secondi.
- Così, mentre una specifica traiettoria balistica veniva calcolata a mano con l'ausilio di una calcolatrice meccanica in circa 12 ore di lavoro, poteva essere prodotta dall'ENIAC in soli 30 secondi.
- In realtà, il primo problema su cui fu impiegato l'ENIAC furono calcoli relativi allo sviluppo della bomba all'idrogeno, su cui stava lavorando in quel periodo John von Neumann (su cui torneremo).
- L'input/output per questo problema aveva richiesto in totale circa un milione di schede perforate.

la storia dell'ENIAC

- Chiaramente, l'ENIAC non era un computer a programma memorizzato, ma partire dal 1948 furono apportate alcune modifiche.
- Le function tables furono utilizzate come una sorta di EPROM per memorizzare (porzioni di) programmi di uso frequente. Era possibile rappresentare 99 istruzioni diverse e un totale massimo di 600 istruzioni per ogni function table
- John von Neumann suggerì di implementare operazioni ad un operando (oggi parleremmo di one-address architecture), in cui:
 - 3 cifre dell'accumulatore 6 erano usate per il program counter (quindi, al massimo 1000 istruzioni), l'accumulatore 8 conteneva “l’indirizzo” del dato da leggere da un’altra function table, e l’accumulatore 15 avrebbe contenuto il valore letto.

la storia dell'ENIAC

- Questa modifica rallentava di 6 volte la velocità di calcolo dell'ENIAC, ma riduceva il tempo necessario alla riprogrammazione della macchina per un nuovo problema a ore invece che giorni.
- Il 16 settembre 1948 Adele Goldstine (moglie di Herman Goldstine) fece girare il primo programma memorizzato nell'ENIAC (scritto da lei stessa per conto di von Neumann)
- Nel 1953 all'ENIAC fu aggiunta una espansione di memoria che permetteva di memorizzare 100 numeri in più rispetto a quelli memorizzabili negli accumulatori.
- L'ENIAC fu spento definitivamente alle ore 23:45 del 2 ottobre 1955, dopo circa 10 anni di funzionamento.

Nota di colore: i computer del futuro

- Si trova spesso in rete un riferimento ad una predizione terribilmente sbagliata che sarebbe stata fatta nel marzo del 1949 in un articolo dal titolo “*Brain that Clicks*” comparso su **Popular Mechanics**, una rivista molto diffusa negli Stati Uniti in quegli anni :
“computer in the future may weigh no more tha 1,5 tons”
- L’articolo si basava sul fatto che l’ENIAC pesava circa 30 tonnellate, e in effetti l’IBM 608 del 1957 sarebbe poi pesato 1,2 tonnellate.
- Il transistor era appena stato inventato, e sarebbe stato ben difficile predirne, nel 1949, le capacità di miniaturizzazione raggiunte negli anni ’70...



Nota di colore: i computer del futuro

- E sempre in rete potreste trovare un'altra citazione, sbagliata nel contenuto e anche erroneamente attribuita a **Thomas Watson Senior**, allora presidente della IBM, che nel 1943 avrebbe dichiarato:
“I think there is a world market for maybe 5 computers”...
- In realtà si trattava del 1953 e del figlio, **Thomas J. Watson Junior**, il quale si riferiva specificamente al primo computer costruito dalla IBM, il 701, che era appena stato presentato a 20 diversi potenziali acquirenti.
- La IBM si aspettava di vendere 5 esemplari del 701, ma alla fine ricevette ordini per 18 modelli (nella sua autobiografia Watson ne da una ulteriore versione leggermente diversa, dove 10 potenziali acquirenti si trasformarono poi in una ventina di ordini)

I primi computer a programma memorizzato: l'EDVAC e l'SSEM

- L'inizio della costruzione dell'ENIAC nel 1943 e la fretta di finirlo per impiegarlo nella guerra congela possibili innovazioni e nuove idee che avrebbero potuto migliorarne la facilità d'uso e le prestazioni
- Da subito Ecker e Mauchly incominciano a lavorare ad un nuovo progetto che chiameranno **Electronic Discrete Variable Automatic Computer**, o **EDVAC**.
- Nel 1944 Eckert scrive la descrizione di un nuovo tipo di memoria la **Mercury Delay Line (MDL)** che dovrebbe avere lo scopo di memorizzare sia dati che il programma che li deve elaborare.
- La Mercury Delay Line era una sorta di memoria dinamica, come le DRAM, *ma ad accesso sequenziale*, in cui i bit erano memorizzati sotto forma di impulsi sonori all'interno di un tubo pieno di mercurio (Turing aveva suggerito una volta di usare del gin...)

L'EDVAC e John von Neumann

- Vari meeting venivano tenuti regolarmente alla Moore School in modo che i ricercatori potessero tenersi aggiornati sui reciproci lavori, e in quegli anni circa la metà erano tenuti da persone che lavoravano all'ENIAC.
- A questi meeting partecipava spesso il matematico **John von Neumann**, in visita alla Moore come consulente.
- Von Neumann scrisse un famoso/famigerato report dal titolo: **First Draft of a Report on the EDVAC**, che contiene la prima descrizione pubblicata della struttura logica di un computer che fa uso del concetto di programma memorizzato (nella memoria del computer)
- È dal nome dell'autore di questo report che deriva l'espressione divenuta di uso comune in informatica **Architettura von Neumann**

L'EDVAC e John von Neumann

- Distribuito da Goldstine a partire dal 25 giugno del 1945 (ma datato 30 giugno), questo documento sollevò in realtà molte polemiche.
- Innanzi tutto, suscitò subito molto interesse, ed in breve tempo varie copie del documento furono inviate a vari centri di ricerca in giro per il mondo (e in cui nasceranno progetti ispirati all'EDVAC).
- Purtroppo, la distribuzione al pubblico di informazioni sull'EDVAC fece sì che l'EDVAC stesso non potesse più essere brevettato.
- Ma poi il documento, a sola firma di von Neumann, conteneva le idee su un computer a programma memorizzato che erano state sviluppate anche da altri (Ecker, Mauchly e Goldstine in particolare) e ampiamente discusse in vari meeting aperti.
- Anzi, per alcuni l'unico merito di von Neumann sarebbe stato quello di tradurre quelle idee in un formalismo logico, di cui era un esperto.
97

L'EDVAC e John von Neumann

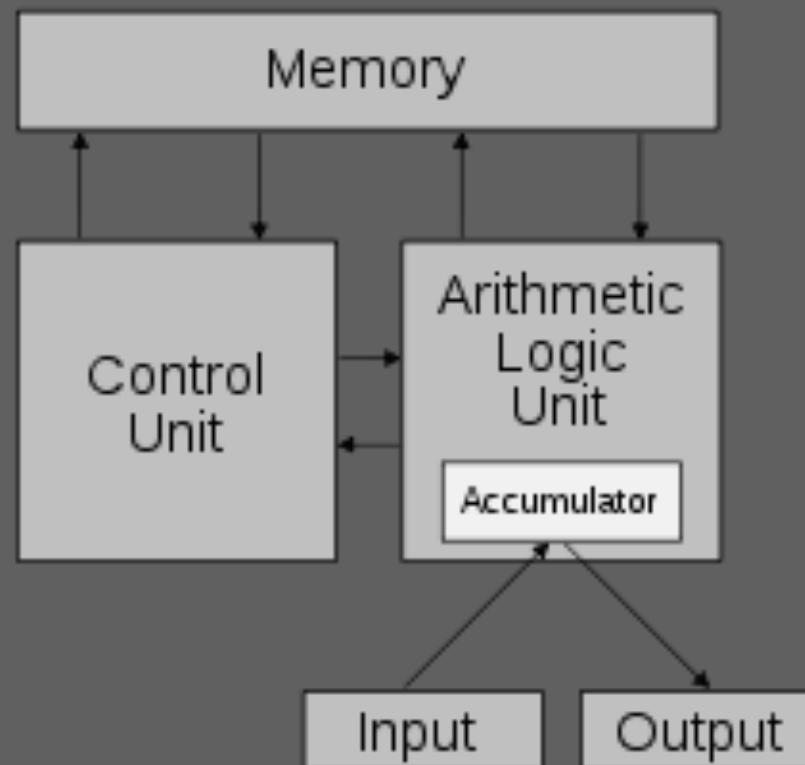
- In una intervista del 2002, Goldstine ribadisce che era stato lui a scrivere il report (lungo 101 pagine), aggiungendo anche alcuni nuovi dettagli tecnici, dato che von Neumann non ne avrebbe avuto il tempo essendo sempre in viaggio.
- In proposito, l'intervistatore osserva che:
 - 1) era noto che von Neumann non amasse scrivere, e lo lasciasse fare ad altri a nome suo;
 - 2) lo stile di scrittura del report è simile ad altri scritti di Goldstine
 - 3) la più vecchia copia del report è scritta a mano con la stessa calligrafia con cui Goldstine scriveva le lettere a von Neumann

L'EDVAC e John von Neumann

- Il fatto è che in quei tempi von Neumann era più anziano e molto più famoso degli altri ricercatori, essendo considerato già dagli anni ‘20 una delle grandi menti matematiche del secolo.
- Goldstine, aveva già lavorato come assistente universitario, per cui gli sembrò normale che i report sui lavori di un gruppo (incluso quello scritto da lui) fossero firmati col solo nome del “barone” del gruppo (Goldstine in realtà usa il termine “senior researcher”)
- Alla luce di questi eventi, è probabile che invece che di “Architettura von Neumann” dovremmo più correttamente parlare di **“Architettura Eckert-Goldstine-Mauchly-von-Neumann”**
- Purtroppo, la storia della scienza è piena di brillanti scienziati i cui meriti non sono stati adeguatamente riconosciuti... 99

L'EDVAC e John von Neumann

- L'architettura von Neumann, o se preferite,
l'architettura Eckert-Goldstine-Mauchly-von-Neumann



L'EDVAC

- Di chiunque fosse stata l'intuizione originale (e probabilmente non lo sapremo mai), l'idea di memorizzare i programmi in una memoria, e potenzialmente nella stessa memoria che contiene i dati è ottima:
 - Rende più veloce l'esecuzione dei programmi
 - Rende più facile e veloce cambiare il programma da eseguire o ricompilarlo dopo averlo modificato
 - Rende più facile (e quindi efficiente) l'esecuzione delle istruzioni di controllo
- I problemi stanno nella necessità di una memoria sufficientemente spaziosa ed efficiente per contenere dati e programmi (e per la tecnologia di quegli anni questo era un vero problema), e il rischio di modificare inavvertitamente il programma stesso.

L'EDVAC

- La costruzione dell'EDVAC inizia nel 1946 e termina nel 1949, ma per una serie di problemi diventa operativo solo nel 1951, con un costo finale di circa 500.000 \$ del tempo (ossia un costo simile all'ENIAC) sempre su commissione dell'esercito americano
- Era composto da quasi 6.000 triodi e 12.000 diodi, per un peso totale di 7 tonnellate e un consumo di 52 KW
- La memoria era composta da 128 mercury delay lines ognuna in grado di memorizzare 8 word da 44 bit (quindi in totale circa 45 Kbit).
- Tre registri erano in grado di memorizzare ciascuno una singola word.
- Una unità computazionale (una ALU) eseguiva operazioni aritmetiche su coppie di numeri e inviava il risultato in memoria.

L'EDVAC

- Al contrario dell'ENIAC, l'EDVAC lavorava effettivamente rappresentando l'informazione in codice binario.
- Ogni istruzione da 44 bit (notate la lunghezza fissa, e la posizione fissa degli operandi, ci torneremo più avanti quando parleremo delle architetture RISC) era suddivisa in 5 campi:
 - 4 bit per il codice dell'istruzione da eseguire
 - 10 bit \times 3 per l'indirizzo dove reperire i due operandi in memoria e l'indirizzo dove scrivere il risultato dell'istruzione
 - 10 bit per l'indirizzo della successiva istruzione da eseguire
- Erano in tutto disponibili solo 12 istruzioni macchina: le 4 operazioni aritmetiche (più moltiplicazione/divisione con arrotondamento), salto condizionato, shift, lettura/scrittura sul nastro perforato, lettura da console di un dato, stop.

L'EDVAC

- Un clock ad 1 MHZ permetteva la lettura di una word in memoria in 48 μ sec, un μ sec per ogni bit di cui era composta (44 bit di informazione + 4 bit a 0 che la separavano dalla word successiva)
- Somme e sottrazioni venivano eseguite in circa 900 μ sec, mentre moltiplicazioni e divisioni in circa 2800 μ sec.
- Notate che non esisteva ancora un concetto vero e proprio di memoria secondaria permanente: le Delay lines dovevano sempre essere tenute operative, oppure i dati dovevano essere salvati su nastro perforato e poi reimmessi in memoria alla successiva accensione.
- L'EDVAC rimase operativo per circa 10 anni, durante i quali furono apportati alcuni miglioramenti, tra cui una memoria a tamburo magnetico (su cui torneremo) in grado di memorizzare altre 4000 word, un lettore e perforatore di schede perforate, un nastro magnetico (torneremo anche su questo), e una unità floating point.

L'eredità dell'EDVAC

- Il progetto dell'EDVAC è sicuramente il primo a contenere l'idea di un computer a programma memorizzato.
- Ma dato che l'EDVAC, diviene operativo solo nel 1951, non si può considerare il primo computer *operativo* a programma memorizzato
- Nell'estate 1946 si tiene alla Moore School un corso intitolato *Theory and techniques for the design of electronic digital computers* sostanzialmente dedicato alla descrizione del progetto EDVAC
- A questo corso partecipano 28 scienziati provenienti da vari centri di ricerca americani e non, e che daranno origine a varie ricerche, progetti e realizzazioni parallele (se mai ci fosse bisogno di dimostrare l'importanza della libera circolazione delle idee...)

L'eredità dell'EDVAC

- Ecco i calcolatori realizzati dal 1946 fino al “debutto” dell’EDVAC
 - **SSEM**, University of Manchester (G.B.) (prototipo: giugno 1948);
 - **EDSAC**, Cambridge University (G.B.) (maggio 1949);
 - **BINAC**, Eckert-Mauchly Computer Corporation (estate 1949);
 - **Whirlwind**, Massachusetts Institute of Technology (inizio 1950);
 - **ACE**, National Physical Laboratory (G.B.) (maggio 1950);
 - **SEAC**, National Bureau of Standards (maggio 1950);
 - **SWAC**, National Bureau of Standards (luglio 1950);
 - **UNIVAC**, Remington Rand Corporation (marzo 1951);
 - **IAS**, Institute for Advanced Study (giugno 1951);

L'eredità dell'EDVAC

- Una costante di queste realizzazioni fu la sperimentazione per mettere a punto una memoria adeguata alle necessità dei circuiti logici che dovevano manipolarne il contenuto.
- Le tecniche per realizzare circuiti elettronici in grado di effettuare operazioni e controlli erano ormai note e consolidate, ma per funzionare adeguatamente dovevano essere collegati ad una memoria elettronica sufficientemente capace, spaziosa ed efficiente.
- Insomma, fu chiaro fin da subito che l'efficienza dei computer non dipendeva solo dalla velocità con cui eseguivano i calcoli, ma dalla velocità con cui era possibile accedere alle istruzioni e ai dati su cui le istruzioni dovevano operare.

L'eredità dell'EDVAC

- Notevole come, in un articolo dal titolo *Preliminary Discussion of the Logical Design of an Electronic Computing Instrument*. [Princeton: Institute for Advanced Studies, September 1947, authors **A. Burks, H. Goldstine, J. von Neumann**] troviamo scritto:

Ideally, one would desire an indefinitely large memory capacity such that any particular...word would be immediately available...We are forced to recognize the possibility of constructing a hierarchy of memories, each which has greater capacity than the preceding but which is less quickly accessible

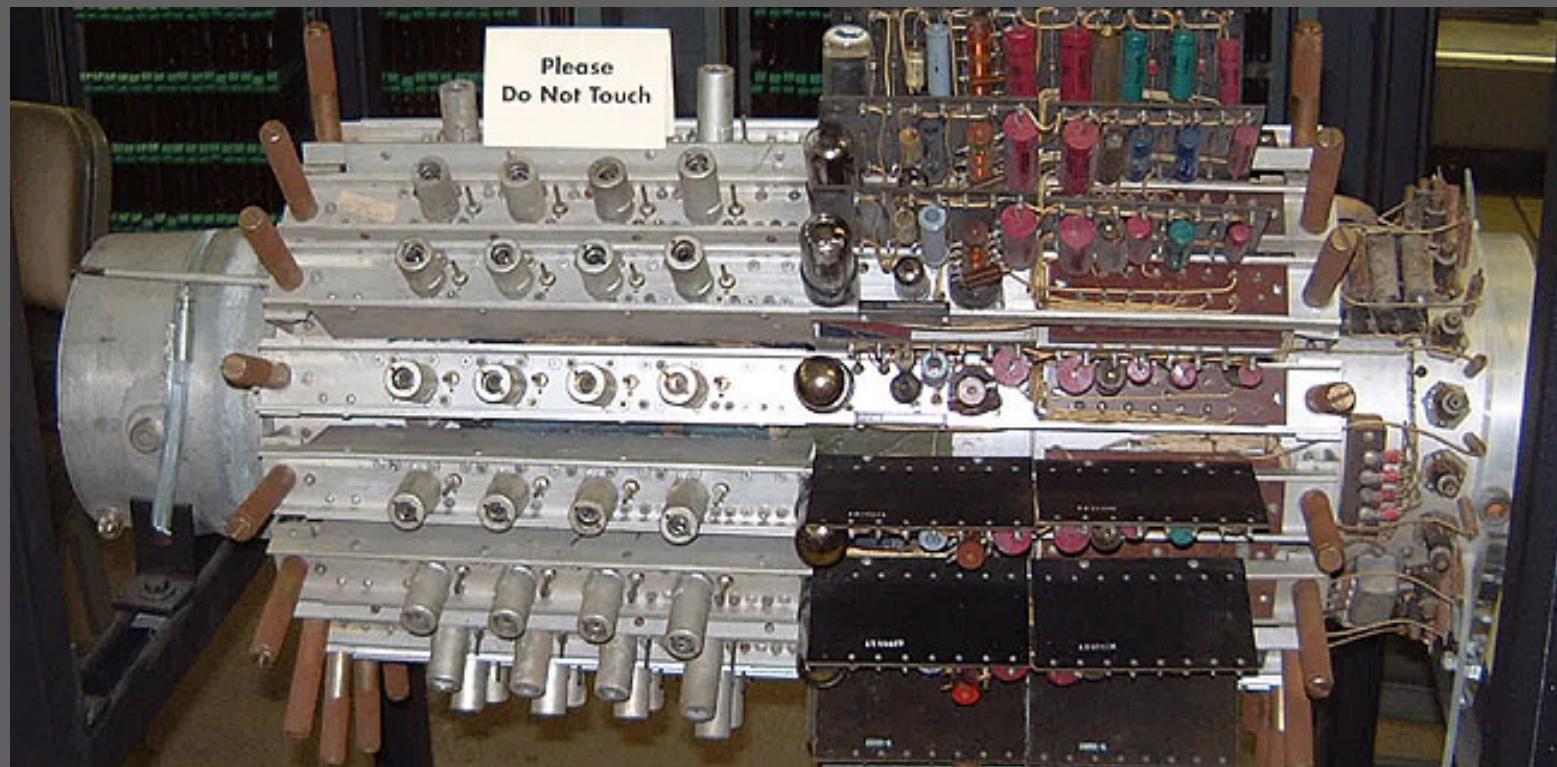
- Quando questi concetti avrebbero richiesto decenni per essere effettivamente realizzati praticamente (nota di colore: una copia originale dell'articolo è stata venduta all'asta da Christies nel 2013 per 4.375 \$)

Manchester Small Scale Experim. Machine

- Altrimenti noto come **SSEM**, operativo dal 21 giugno 1948 è il computer a cui spetta la palma di **primo computer a programma memorizzato**.
- l'SSEM fu progettato da **F. Williams e T. Kilburn** inizialmente solo come piattaforma per testare il funzionamento del **Williams-Kilburn Tube (WKT)**, un tipo di memoria per computer che eliminava alcuni dei difetti delle Mercury Delay Lines.
- Le MDL infatti erano memorie ad accesso sequenziale. Nell'EDVAC una “linea” conteneva 8 word (di 44+4 bit ciascuna) che potevano essere lette solo in sequenza.
- Inoltre le MDL potevano funzionare solo in condizioni di temperatura estremamente controllate, ed erano molto ingombranti.

Manchester Small Scale Experim. Machine

- Qui vediamo la memoria dell'UNIVAC, in grado di memorizzare in tutto 1000 word da 40 bit ciascuna. Ogni mercury line contiene 120 bit, ossia 3 word, che devono essere lette in sequenza per accedere alla word indirizzata.



Manchester Small Scale Experim. Machine

- Il Williams-Kilburn Tube si può invece considerare la prima vera e propria memoria ad accesso diretto, o RAM.
- Poiché necessitava di un refreshing periodico dell'informazione memorizzata, era più propriamente una DRAM (proprio come le Mercury Delay Lines).
- Consisteva in un tubo catodico (CRT) del diametro di 15-30 cm, la cui superficie era suddivisa in una matrice rettangolare di 1024–2560 slot, ognuno dei quali poteva essere caricato (ossia scritto) positivamente o negativamente con un opportuno raggio sparato in direzione dello slot stesso.
- Rispetto alle MDL, i WKTs occupavano molto meno spazio, ed erano ad accesso effettivamente diretto, oltre che molto più veloce (per contro tendevano ad essere meno affidabili con il tempo).

Manchester Small Scale Experim. Machine

- Il SSEM era una macchina a 32 bit (proprio come le prime macchine RISC progettate 30 anni più tardi, e su cui torneremo) che pesava “solo” una tonnellata, impiegava circa 1000 tubi a vuoto, consumava 3,5 KW e fu assemblata in soli 6 mesi.
- La ALU eseguiva solo somme, sottrazioni, e negazioni, tutte le altre operazioni venivano fatte via software.
- Usava 4 WKTs per implementare:
 - una memoria RAM da 32 word da 32 bit ciascuna
 - un registro a 32 bit contenente l’istruzione in esecuzione
 - un registro a 32 bit usato come accumulatore (risultato della ALU)
 - una sorta di monitor per visualizzare il contenuto degli altri CRT
- Ogni word della RAM conteneva un dato, o una istruzione di formato:



Manchester Small Scale Experim. Machine

- Le istruzioni erano ad un solo operando, con indirizzamento implicito del secondo operando, costituito dal contenuto dell'accumulatore:

Bin code	Modern mnemonic	semantic
000	JMP S	Jump to the instruction at the address obtained from the specified memory address S (absolute unconditional jump)
001	JRP S	Jump to the instruction at the program counter plus (+) the relative value obtained from the specified memory address S ^[t 1] (relative unconditional jump)
010	LDN S	Take the number from the specified memory address S, negate it, and load it into the accumulator
011	STO S	Store the number in the accumulator to the specified memory address S
100	SUB S	Subtract the number at the specified memory address S from the value in accumulator, and store the result in the accumulator
110	CMP	Skip next instruction if accumulator contains a negative value
111	STP	Stop

Manchester Small Scale Experim. Machine

- Si può facilmente mostrare che queste istruzioni erano sufficienti a rendere il SSEM Turing completo (esercizio: come si poteva eseguire la somma di due numeri?)

LDN X // load negative X into the accumulator

SUB Y // subtract Y from the value in the accumulator

STO S // store the result at S

LDN S // load negative value at S into the accumulator



$$x+y = -(-x-y)$$

- La macchina non aveva un vero e proprio dispositivo di input: ogni word della memoria (che contenesse istruzioni o dati) veniva inizializzata usando 32 switch che permettevano di stabilire il valore di ogni singolo bit (ossia, i programmi erano veramente scritti in binario, direttamente in RAM! Ma del resto, con 32 word disponibili, erano comunque programmi molto corti...)
- La lettura/scrittura di una word in memoria richiedeva 360 μ sec, e il SSEM eseguiva circa 700 istruzioni al secondo.

Manchester Small Scale Experim. Machine

- Furono scritti solo tre programmi per il SSEM!
- Il primo, scritto da Kilburn, era formato da 17 istruzioni, e cercava il più grande fattore primo di 2^{18} testando tutti i numeri da $2^{18}-1$ in giù.
- Fu fatto girare il 21 giugno 1948 eseguendo circa 2,2 milioni di istruzioni e circa 3,5 milioni di accessi in memoria, in 52 minuti.
- Il terzo programma fu scritto da Alan Turing, che in quel periodo insegnava al dipartimento di matematica dell'università di Manchester, per eseguire divisioni su numeri molto grandi.
- Una descrizione del SSEM fu pubblicata da Williams e Kilburn sulla rivista **Nature** nel settembre del 1948

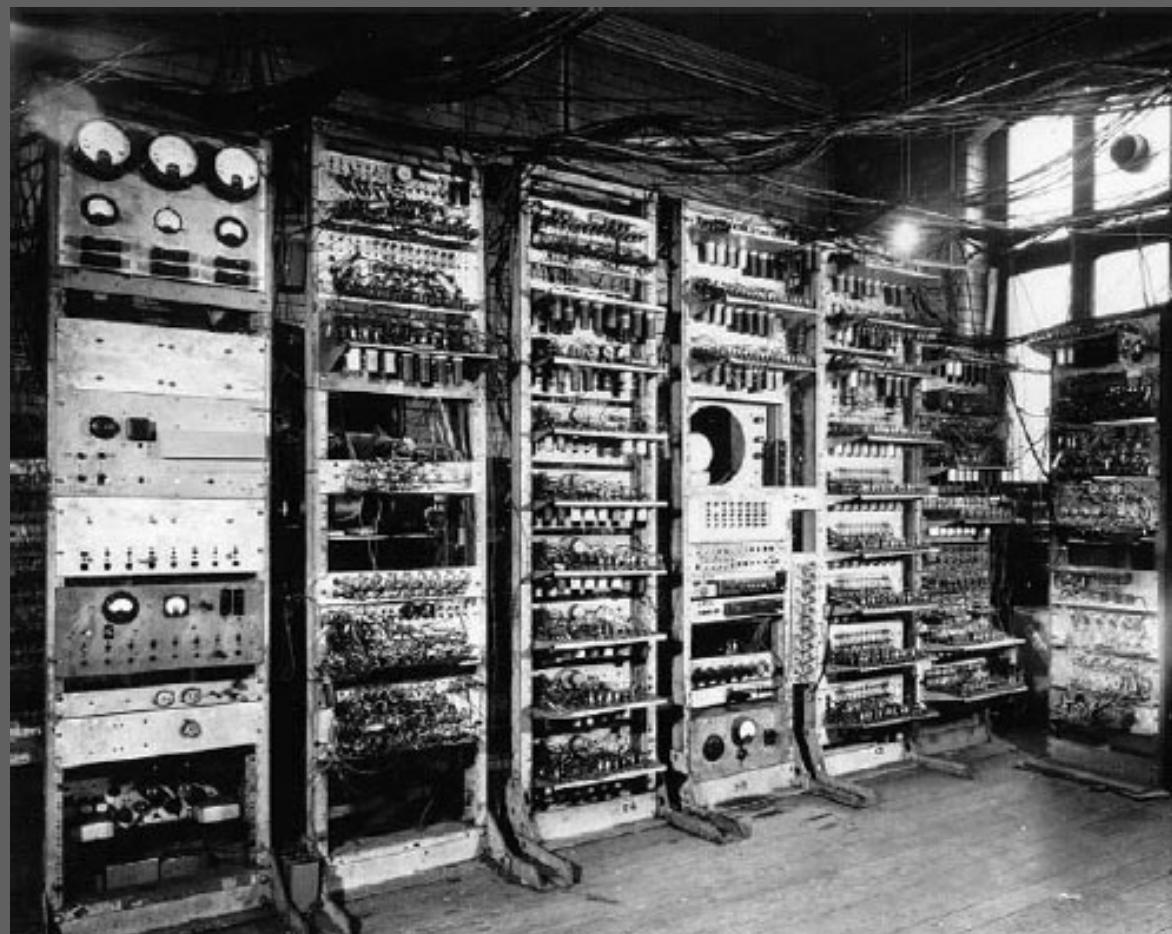
Manchester Small Scale Experim. Machine

1967/49 Kilburn Highest Factor Routine (amended)

Instruction	C	25	26	27	Line	01	2	3	4	5	13	14	15
-245 C	-G ₁	-	-	-	1	000	011	010					
+ 26			-G ₁		2	010	011	110					
-265 C	G ₁		-G ₁	G ₁	3	010	011	010					
+ G 27			-G ₁	G ₁	4	110	011	110					
-235 C	a	T ₀₀₁	-G ₁	G ₁	5	111	001	010					
Sub. 27	a=ab				6	110	011	001					
Test					7	—	011						
Add 205 C					8	000	001	100					
Sub. 26	T ₀				9	010	111	001					
+ G 25	T ₀				10	100	111	110					
-265 C					11	100	011	010					
Test					12	—	011						
Stop	0	0	-G ₁	G ₁	13	111							
-265 C	G ₁	T ₀	-G ₁	G ₁	14	010	111	010					
Sub. 21	G ₁				15	101	011	001					
+ G 27	G ₁				16	110	111	110					
-275 C	G ₁				17	110	111	010					
+ G 26	G ₁				18	010	111	110					
-225 C	T ₀		-G ₁	G ₁	19	011	011	000					
	20	-3	10111	etc.	23	-a			int.	final			
	21	1	10000		24	b ₁			25	—	T ₀₍₀₎		
	22	4	00100						26	—	-G ₁		
									27	—	G ₁		
											or 10100		

Il programma scritto da Kilburn per il SSEM

Il SSEM (notate al centro, rotondo, il tubo catodico per leggere il contenuto delle tre memorie WKT del SSEM)



Manchester Small Scale Experim. Machine

- Il successo del SSEM e del nuovo tipo di memoria sperimentato, portò rapidamente alla progettazione e costruzione di un computer più potente e di più facile uso, il **Manchester Mark 1**, che cominciò a funzionare nell'aprile del 1949.
- Alla Ferranti, una compagnia inglese che produceva componenti elettrici ed elettronici, fu commissionata la produzione e la commercializzazione del Mark 1 con il brand **Ferranti Mark 1**, che a partire dal 1951 divenne così il primo computer della storia ad essere disponibile sul mercato (almeno 7 modelli venduti dal '51 al '57).
- A proposito degli esperimenti col SSEM, alcuni anni più tardi Williams scrisse: *A program was laboriously inserted and the start switch pressed. [...] it stopped, and there, shining brightly in the expected place, was the expected answer. It was a moment to remember. This was in June 1948, and nothing was ever the same again.*¹¹⁷

Il Whirlwind I del MIT

- Alcuni considerano il **Whirlwind I**, sviluppato al **Massachusetts Institute of Technology (MIT)**, il primo computer a programma memorizzato della storia.
- In effetti il progetto di questo computer da parte di **Jay Forrester** e **Robert Everett** incomincia già nel 1947, ma la macchina verrà realizzata fisicamente solo nell'aprile del 1951.
- In ogni caso il Whirlwind conteneva due fondamentali innovazioni che verranno poi adottate da tutti i progetti degli anni a venire (con poche eccezioni).
- I computer realizzati fino a quel momento operavano in **bit-serial mode**, cioè i bit che componevano le word di dati venivano elaborati in serie, uno dopo l'altro, dalla logica del computer.
- Questo perché nelle memorie MDL e WKT si poteva leggere un solo bit alla volta.

Il Whirlwind I del MIT

- Il Whirlwind invece fu il primo computer ad operare in parallelo sui 16 bit che componevano le sue word, usando opportune unità logiche in parallelo.
- Questo naturalmente richiedeva una maggior quantità di hardware, ma rendeva in principio il Whirlwind 16 volte più veloce di qualsiasi altro computer, a parità di tutte le altre condizioni.
- Ma tra le “altre condizioni” c’era la necessità di usare memorie che permettessero di accedere ai bit di ciascuna word ad una velocità simile a quella con cui il Whirlwind li elaborava.
- MDL e WKT erano troppo lente, e dopo una serie di ricerche ed esperimenti Forrester brevettò le **core memory**, o **memorie a nucleo magnetico**, che verranno usate nel Whirlwind e in molti altri computer degli anni successivi (e su cui torneremo più avanti). 119

Il Whirlwind I del MIT

- La seconda innovazione fondamentale riguarda i concetti di **microcodice** e **micropogramma**, quel livello intermedio tra le porte logiche di un processore e le sue istruzioni macchina.
- Nel Whirlwind era per la prima volta presente un **control store** (in sostanza una ROM costruita mediante una matrice di diodi) che stabiliva, per ogni istruzione, come pilotare le porte logiche necessarie ad eseguire una certa istruzione
- Ad ogni ciclo di un “master clock”, una linea di segnali all’interno della matrice di diodi veniva selezionata, in modo da pilotare opportunamente i vari circuiti logici della macchina (ad esempio i multiplexer e la ALU) necessari all’esecuzione dell’istruzione corrente.

Il Whirlwind I e la microprogrammazione

- L'idea fu poi sviluppata a pieno da **Maurice Wilkes**, nel 1951. La matrice di diodi del Whirlwind poteva essere vista come una memoria di **microistruzioni**, una per ogni riga della matrice stessa.
- Wilkes aggiunse una seconda matrice che permetteva di decidere quale fosse la prossima microistruzione da eseguire (ossia quale fosse la prossima riga di segnali da selezionare nella prima matrice), sulla base dell'esito dell'applicazione della precedente microistruzione.
- In sostanza quindi, con la seconda matrice venivano introdotte le **microistruzioni condizionali**, le quali permettevano di eseguire sequenze diverse di microistruzioni a seconda dell'istruzione macchina in esecuzione. (in figura l'unità di controllo del Whirlwind, con a sinistra la core memory)



Il Whirlwind I e la microprogrammazione

- L'esecuzione di una istruzione macchina poteva quindi essere scomposta nell'esecuzione di una sequenza ben determinata di microistruzioni selezionate dalla matrice di microistruzioni usando la seconda matrice.
- Wilkes chiamò queste sequenze **microprogrammi**, e il concetto corrispondente **microprogrammazione**: appunto uno strato intermedio di microistruzioni tra le porte logiche della macchina ed il suo instruction set.
- In principio, cambiando le due matrici si poteva dotare la macchina di un diverso set di istruzioni, o arricchire quello esistente, allo stesso tempo rendendo più semplice la progettazione della CPU. Una tecnica che verrà ampiamente sfruttate negli anni successivi.

Pionieri dei primi computer...

- Il periodo che va dal 1946 al 1952 fu quindi estremamente fertile per lo sviluppo dei computer come li conosciamo oggi. La disseminazione delle idee in conferenze e articoli scientifici, e i prototipi costruiti nei centri di ricerca permisero di sperimentare con diverse idee, diversi modelli architetturali e diverse soluzioni tecnologiche.
- Alcune soluzioni si rivelarono migliori di altre, e su di esse si concentrarono i ricercatori per migliorarle e raffinarle, dando origine ad un corpus iniziale di conoscenze sulle architetture dei computer che andavano consolidandosi e venivano adottate per i nuovi progetti.
- In definitiva, gli anni ‘50 divennero un periodo pieno di innovazioni tecnologiche e processi produttivi, ma anche pieno di idee originali nel campo del software: linguaggi di programmazione e sistemi operativi (di cui però parleremo nelle rispettive sezioni).

... e primi impatti culturali

- E in quegli anni echi degli esperimenti con i primi computer cominciarono anche a raggiungere il grande pubblico.
- Il SSEM e il Mark 1 furono oggetto di vari articoli da parte della stampa britannica, che usò spesso, riferendosi a questi computer, l'espressione “electronic brain”.
- Il “Time” ospitò un dibattito tra sostenitori e detrattori dell’idea che i computer potessero in qualche maniera replicare il funzionamento del cervello umano (o per lo meno le sue capacità).
- In uno di questi articoli, un intervista a Turing, egli osservò:

This is only a foretaste of what is to come, and only the shadow of what is going to be. We have to have some experience with the machine before we really know its capabilities. It may take years before we settle down to the new possibilities, but I do not see why it should not enter any of the fields normally covered by the human intellect and eventually compete on equal terms

... e primi impatti culturali

- Nel 1952 alla televisione, viene usato un UNIVAC 1 per predire in diretta i risultati delle elezioni presidenziali.
- L'apparizione in TV di un computer espande la consapevolezza del grande pubblico americano nei confronti dei computer.
- In figura, la copertina di un numero di Superman di quegli anni, in cui un UNIVAC viene usato per cercare marito a Lois Lane.



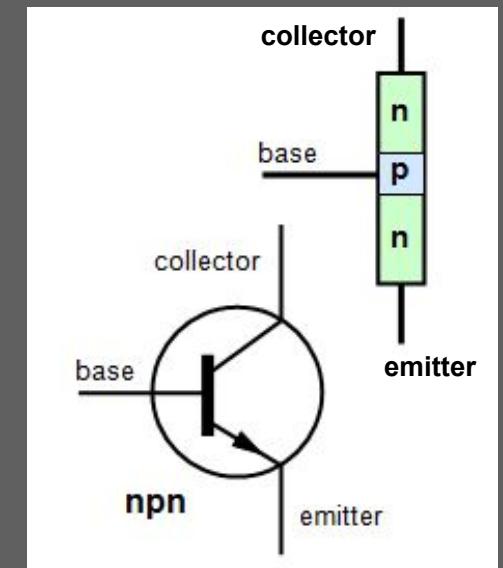
Transistors a semiconduttori: the stuff modern computers are made of

- Il 23 dicembre del 1947, ai laboratori **Bell**, i fisici **John Bardeen**, **Walter Brattain** e **William Shockley** annunciano lo sviluppo del primo *transconductance-varistor*, meglio noto come **transistor** (Per questa scoperta riceveranno il premio Nobel per la fisica nel 1956)
- La tecnologia di costruzione verrà più volte migliorata, fino a divenire quella che più o meno conosciamo, basata sull'uso di materiale semiconduttore (silicio in particolare)
- I transistor sostituiscono le valvole termoioniche migliorandole in molti modi: costano meno, consumano meno, scaldano meno, si rompono di meno, e sono molto più piccoli (e col tempo si scoprirà come miniaturizzarli fino a dimensioni microscopiche)



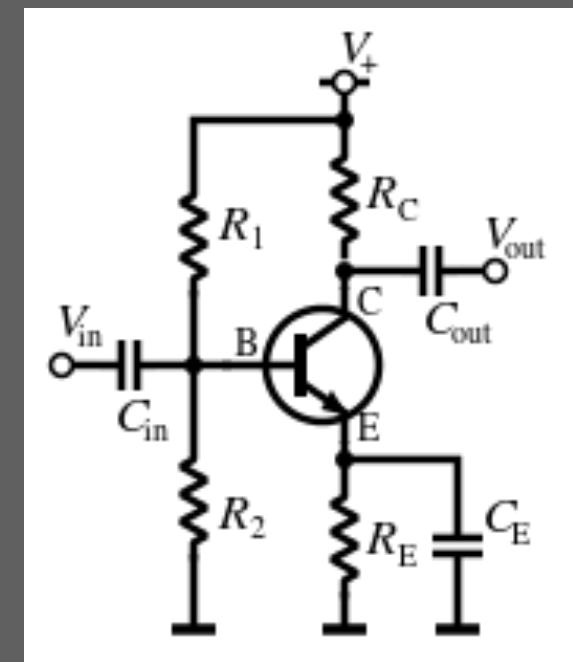
Il transistor

- Un semiconduttore (come il silicio, il germanio, l'arseniuro di gallio) è materiale con una condutività intermedia tra un materiale conduttore e uno non conduttore (di corrente elettrica)
- Il semiconduttore può essere drogato con cariche positive (P) o negative (N), in questo modo alterando le sue proprietà elettriche.
- Un transistor è composto da tre strati di materiale semiconduttore, opportunamente drogati in modo alternato (ad esempio NPN) e sovrapposti)
- I tre strati, a cui viene applicata una tensione elettrica, prendono il nome di base, collettore ed emettitore (rispettivamente lo strato centrale P e i due strati esterni N)



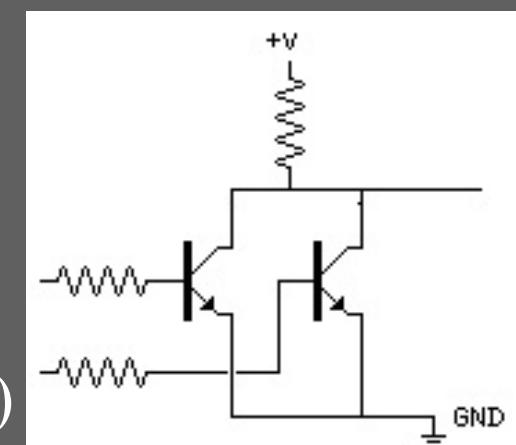
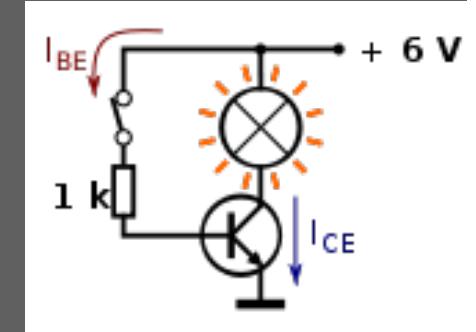
Il transistor

- In maniera simile ad una valvola termoionica, il transistor può essere usato come amplificatore di segnale elettrico.
- Applicando una (piccola) tensione variabile V_{in} alla base, è possibile ottenere una corrispondente variazione (di tensione, di corrente o di entrambe, a seconda del tipo di circuito adottato) del segnale in V_{out} , cioè in uscita al collettore.
- Praticamente qualsiasi dispositivo elettronico utilizza questa proprietà amplificatrice dei transistor.



Il transistor

- Ma un transistor (proprio come un relè o una valvola termoionica) può anche essere usato come interruttore elettrico.
- A seconda che la tensione applicata alla base del transistor sia superiore o inferiore ad una certa soglia, la corrente fluisce o meno tra il collettore e l'emettitore del transistor
- E quando è usato come switch, il transistor fornisce un altro vantaggio fondamentale rispetto alle valvole termoioniche: riesce a commutare molto più velocemente il proprio stato aperto/chiuso.
- Dunque può funzionare a frequenze più elevate (del clock di un computer, ad esempio)
- **Quindi, coi transistor possiamo costruire circuiti logici molto più veloci, affidabili, e compatti, e con consumi minori.** (in figura, porta NOR a transistor: ingressi a sinistra, output a destra)



Computer a transistor

- Ovviamente, i primi transistor non erano compatti e affidabili come quelli usati oggi, e la tecnologia di produzione era ancora sperimentale e incerta, ma la strada era tracciata.
- Un importante innovazione si ha nel 1954, quando la **Texas Instr.** commercializza il primo transistor al silicio
- Nel 1953 **Tom Kilburn** costruisce una versione del Manchester Mark a transistor, che diviene operativo come primo computer a transistor nel novembre del 1953. Nel 1955 ne viene costruita una seconda versione, che usa 200 transistor e consuma circa 125 Watt.
- Seguiranno presto altri modelli di computer a transistor, come semplici prototipi o come prodotti commerciali. I primi furono, sempre nel 1955, l'Harwell CADET (in Gran Bretagna) e l'IBM 608, e nel 1956 l'UNIVAC a transistor.

Memorie e computer negli anni ‘50

- Dopo il transistor, per i computer un’altra importante innovazione tecnologica degli anni 50 riguarda le memorie.
- Le MDL e i WKT infatti si rivelarono presto inadeguate a fornire una sufficiente velocità di accesso e spazio di memorizzazione per le necessità dei computer che venivano man mano sviluppati.
- Negli anni 50 vengono sviluppate nuove tecniche di memorizzazione, i **nastri magnetici**, le **rotating drum memory**, le **magnetic core memory** e, nella seconda metà degli anni ‘50, gli **hard disk**.
- La distinzione per noi normale tra memoria primaria e memoria secondaria doveva però ancora emergere chiaramente. Diversi progetti utilizzavano configurazioni e tecnologie mai sperimentate prima, alcune soluzioni si dimostravano efficienti e venivano mantenute e migliorate nei progetti successivi, altre venivano abbandonate. 131

Memorie e computer negli anni ‘50

- L'**UNIVAC I** (**UNI**versal **A**utomatic **C**omputer **I**) progettato dalla **Eckert-Mauchly Computer Corp.** (poi assorbita dalla Remington Rand), diviene operativo nel marzo del 1951. Uno dei primi computer commerciali, ne vengono venduti in tutto 46 modelli (fino a raggiungere un prezzo intorno a 1,3 milioni di dollari!)
- Come abbbiamo già visto, usava come memoria principale una Mercury Delay Line, mentre come memoria di massa l'UNIVAC è il primo computer ad adottare un **nastro magnetico** (denominato **UNISERVO**)
- I programmi da eseguire venivano memorizzati sul nastro di carta usando direttamente una telescrivente oppure mediante un lettore di schede perforate, collegati al dispositivo che gestiva il nastro magnetico

Memorie e computer negli anni ‘50

- I nastri magnetici ricopriranno un’importanza fondamentale fino agli anni ‘70, come unico supporto di memoria di massa veramente capiente, seppure di dimensioni relativamente ingombranti.
- Col tempo aumenta la velocità di accesso ai dati e la quantità di informazione memorizzata per cm².
- Sebbene in modo molto più marginale vengono ancora usati.
- Nel maggio del 2014 la **Sony** ha annunciato una tecnologia di memorizzazione su nastro con una densità di 148 Giga bit per pollice quadrato (2,54² cm²)



Memorie e computer negli anni ‘50

- Nel 1952 incomincia la commercializzazione dei **Mainframe IBM**, grandi sistemi hardware costruiti a valvole e catalogati **serie 700**.
- Dal 1956 verranno costruiti a transistor, ed indicati come **serie 7000**
- Domineranno il mercato dei computer fino alla metà degli anni 60, quando verranno sostituiti dalla famosissima serie **system/360-370**
- Negli anni ‘50 (e poi negli anni ’60 e ‘70), il termine *mainframe* era diventato quasi sinonimo di *computer IBM*.
- Il termine “**mainframe**” deriva da “**main frames**”, grossi armadi in cui erano ospitati la CPU (i circuiti integrati non esistevano ancora) e la memoria del computer, ma veniva usato normalmente per indicare grossi sistemi, con capacità di elaborazione molto elevate (per quei tempi), costosi e quindi usati principalmente da grandi compagnie e dai governi.

Memorie e computer negli anni ‘50

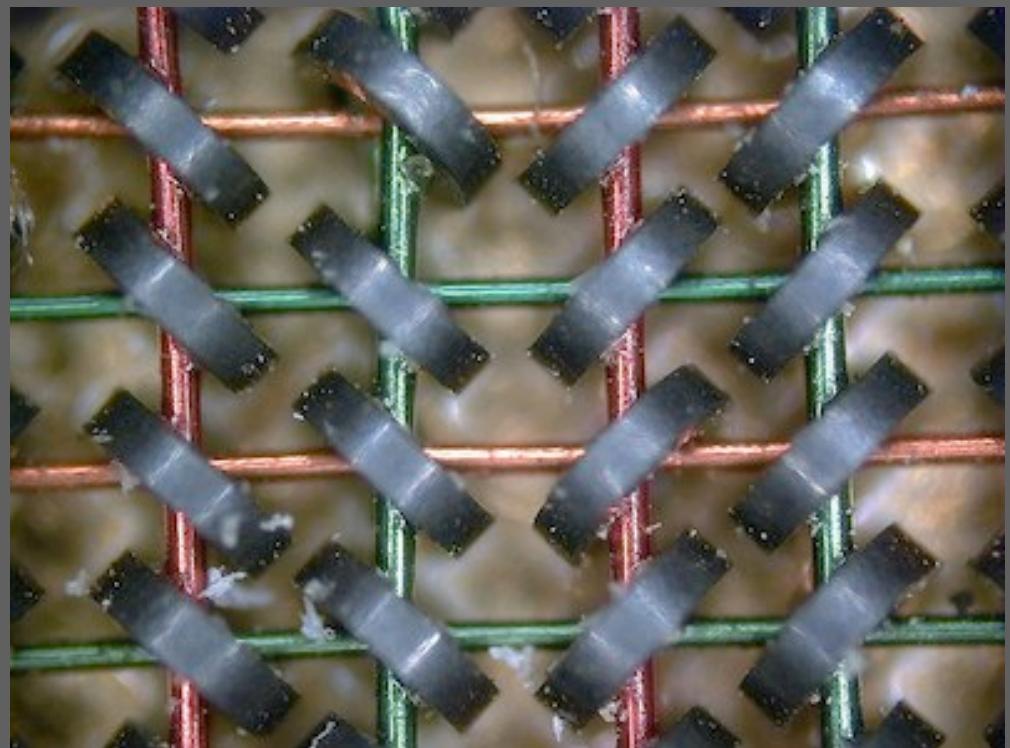
- I mainframe della serie IBM 700/7000 erano suddivisi in due grandi categorie, a seconda del tipo di impiego:
- **Applicazioni scientifico ingegneristiche**: 701, 704, 709, 7090, 7040. Macchine orientate soprattutto ai calcoli interi e floating point, con word a 36 bit (il 7030 usava word a 64 bit)
- **Applicazioni commerciali e elaborazione dati**: 702, 705, 7010, 7080. Macchine orientate alla manipolazione di stringhe.
- I modelli delle due serie usavano periferiche simili (memoria, I/O) **ma avevano set di istruzioni totalmente incompatibili**
- I primi modelli della serie usavano memorie WKT, poi sostituite dalle **memorie a tamburo rotante** e dalle **Magnetic Core Memory (MCM)**. Una tecnologia che dominerà il settore fino a metà degli anni ‘70 (quando verrà sostituita dalle memorie a semiconduttore). ¹³⁵

Memorie a Nucleo Magnetico

- Le memorie a nucleo magnetico sono state la forma predominante di memoria primaria all'incirca dal 1955 al 1975 (anche se spesso venivano usate insieme alle memorie a tamburo rotante)
- Furono il risultato di varie ricerche indipendenti condotte già a partire dagli anni '40, sebbene il primo brevetto a riguardo è del 1951 a nome di **Jay Forrester** (che abbiamo già visto a proposito del Whirlwind)
- La ricerca nasceva dall'esigenza di trovare un sostituto ai WKT, che erano sostanzialmente inaffidabili.
- Le MCM sfruttano la proprietà di alcuni materiali dotati di campo magnetico in cui la direzione del campo cambia se sono sottoposti ad un altro campo magnetico di sufficiente intensità.
- La direzione del campo può quindi essere usata per indicare il valore di un bit di informazione: 0 oppure 1

Memorie a Nucleo Magnetico

- Fili elettrici adeguatamente intrecciati ai nuclei di ferrite (che di solito hanno una forma ad anello) permettono di generare il campo magnetico col quale polarizzare opportunamente ogni singolo nucleo, memorizzando così su di esso un valore 0 oppure 1.
- La polarizzazione della ferrite può essere cambiata, ma si conserva nel tempo, di conseguenza le MCM erano memorie non volatili.
- Un terzo filo passante tra gli anelli (non presente in figura) permetteva la lettura del bit memorizzato, che poi però andava riscritto perché l'anello di ferrite si smagnetizzava.
- Nella figura, la distanza tra gli anelli è di circa 1 mm



Memorie a Nucleo Magnetico

- Le prime MCM avevano tempi di accesso intorno ai 10 µsec, che scesero a meno di un µsec per quelle prodotte negli anni ‘70.
- Benché usate come memoria primaria, erano permanenti, e quindi in definitiva molto simili alle moderne memorie a stato solido.
- Erano tuttavia molto costose: nel 1960 un bit di core memory costava circa 0,2 \$. nel 1980 una MCM da 32 KB costava intorno ai 3.000\$, e aveva una dimensione di circa 25×20 cm.
- L'IBM 704 del 1954 usava una core memory da 4096×36 bit. Una dimensione di memoria primaria che non cambierà di molto nel tempo
- Erano spesso chiamate semplicemente “core memory”, o “core”, ed è per questo che a volte la memoria primaria di un computer viene ancora indicata con quell'espressione. È per la stessa ragione che il file che registra il contenuto della memoria dopo un errore di sistema viene chiamato “**core dump**”.

Il primo computer per le masse: l'IBM 650

- Nel 1953 era però già stato annunciato l'**IBM 650**, che ricopre un ruolo importante nella storia dell'informatica (in realtà le consegne iniziarono a dicembre del 1954)
- Fu il primo computer ad essere commercializzato in grandi quantità (per gli standard dell'epoca) per una ampia gamma di applicazioni.
- Dal 1953 al 1962 ne furono prodotti circa 2000 esemplari, molti di più di tutti i computer della serie 700 messi assieme. In effetti fu il primo computer che permise dei profitti rilevanti alla IBM (la quale si aspettava inizialmente di venderne una cinquantina...)
- I motivi del successo sono che “costava poco” (200.000÷400.000 \$, e veniva affittato per circa 3000\$/mese), dimensioni “ contenute” (stava in una stanza), ed era “facile” da usare dato che aveva un instruction set semplice e veniva programmato in decimale anziché in binario!³⁹

Il primo computer per le masse: l'IBM 650

- L'IBM 650 era veramente un “general purpose computer”, e come tale venne impiegato in università e scuole superiori per l'insegnamento della programmazione, nonché in una ampia gamma di applicazioni scientifiche, ingegneristiche e commerciali.
- Molti lo considerano **l'antenato dei PC**. Infatti, con i computer più grandi e costosi, i programmati non avevano accesso alla macchina, ma dovevano consegnare il loro programma su schede perforate ad un operatore per riaverle indietro qualche giorno dopo insieme all'output
- Con il 650, ai programmati era concesso di usarlo direttamente per uno slot di tempo, lanciare il programma, correggere gli errori e rilanciare il tutto (Certo, a causa del suo costo, il 650 era in servizio 24 ore su 24, così lo slot assegnato poteva essere alle 3 di mattina...)

Il primo computer per le masse: l'IBM 650

- La configurazione base prevedeva solo l'unità principale con la console di controllo, il modulo di alimentazione (dietro, nella foto) e il lettore/perforatore di schede (a sinistra, se ne vede solo metà).
- Poteva però essere esteso con altri prodotti IBM: lettore di nastri magnetici, stampante da schede perforate, e più avanti con hard disk.
- Il 650 funzionava a tubi catodici, con memoria a tamburo rotante (che vedremo più avanti). Per questa ragione era anche noto come **“Magnetic Drum Calculator”**
- Nel 1959 fu commercializzata la versione a transistors (7070), la cui memoria primaria era costituita da una combinazione di memorie a nucleo magnetico e memorie a tamburo rotante.



Il primo computer per le masse: l'IBM 650

- Il 650 elaborava dati in decimale, non in binario. I dati erano memorizzati in word contenenti valori decimali a 10 cifre, e le istruzioni operavano su numeri memorizzati in questo formato.
- I numeri erano rappresentati in forma “**bi-quinaria**”, in cui ci vogliono 6 bit per rappresentare una cifra da 0 a 9:
 - Dei primi 5 bit, uno solo viene messo a 1, per indicare un valore da 0 a 4. Se il sesto bit è a uno, allora al valore indicato dai primi 5 bit va aggiunto 5.

- Ecco il numero 281: 00100 0

00100
└┘

2 + 0

00010 1
└┘

3 + 5

01000 0
└┘

1 + 0

- In effetti, non stupisce che questa notazione non abbia avuto molto successo e sia stata presto abbandonata...

Il primo computer per le masse: l'IBM 650

- Ma appunto, permetteva di scrivere codice usando almeno numeri decimali, anziché in binario. In un periodo in cui i compilatori erano ancora qualcosa di esotico, non era poco...
- Il 650 aveva in tutto 3 registri per contenere:
 - l'istruzione in esecuzione (program/instruction register)
 - il dato indirizzato dall'istruzione in esecuzione (distributor)
 - il risultato dell'esecuzione dell'istruzione (accumulator)
- La memoria a tamburo rotante era composta da 1000, 2000 o 4000 word, e ogni word poteva memorizzare un intero a 10 cifre con segno o 5 caratteri alfanumerici.

Il primo computer per le masse: l'IBM 650

- Le istruzioni macchina avevano un formato fisso: xx yyyy zzzz dove:
- xx: **il codice operativo dell'istruzione da eseguire**
- yyyy: **l'indirizzo in memoria del dato da usare**. Nelle operazioni a due operandi uno era costituito dal contenuto del registro accumulator, che avrebbe poi contenuto il risultato dell'operazione. Era quindi una forma di indirizzamento隐式 (implicito)
- zzzz: **l'indirizzo della prossima istruzione da eseguire**. Questa soluzione era stata adottata per ottimizzare l'accesso alle istruzioni di un programma che devono essere eseguite in una certa sequenza, ma sono memorizzate su un tamburo rotante. Quindi, appena letta una istruzione, conviene che la prossima istruzione da eseguire non sia memorizzata nella word immediatamente successiva (che è appena stata superata dalla testina di lettura) ma un po' più in là... 144

Il primo computer per le masse: l'IBM 650

- Un'istruzione macchina aveva quindi l'aspetto di un numero decimale a 10 cifre come, ad esempio, 1905400345:
 - moltiplica il numero memorizzato alla word di indirizzo 0540 con il contenuto dell'accumulator e metti il risultato nell'accumulator;
 - vai ad eseguire l'istruzione memorizzata alla word 0345
- Possiamo immaginare quanto fosse scomodo scrivere programmi come sequenze di numeri a 10 cifre, dovendo anche decidere a quale indirizzo memorizzare ogni word (e come organizzare le istruzioni se si voleva ottimizzare il tempo di esecuzione)
- Comunque, sempre meglio che scrivere i programmi in binario...
- Il 650 eseguiva la maggior parte delle istruzioni in 0,4 msec, ma una moltiplicazione richiedeva 10 msec, e una divisione 15 msec

Il primo computer per le masse: l'IBM 650

- In realtà, già dal 1955-1956 fu possibile scrivere i programmi in assembler, in modo almeno da non dover ricordare a memoria i codici operativi e tenere traccia di dove stavano i dati in memoria.
- Dal 1957 fu finalmente disponibile un compilatore per il FORTRAN, il primo linguaggio di programmazione “machine independent”
- Col tempo furono sviluppati altri linguaggi (e relativi compilatori) per il 650. Tra cui: ADES II, BACAIC, BALITAC, BELL, CASE, SOAP III, COMTRAN, DRUCO I, DYANA, EASE II, ELI, ESCAPE, FAST, FLAIR, GAT, IPL, IT, KISS, L1, L2, MAC, MITILAC, MYSTIC, OMNICODE, RELATIVE, RUNCIBLE, FORTRUNCIBLE, SIR, SPEEDCODING, SPIT, SPUR.
- Si, tutti scomparsi. Come del resto le architetture e i sistemi operativi, anche i linguaggi di programmazione sono stati sottoposti ad una vera e propria selezione naturale...
140

Il primo computer per le masse: l'IBM 650

- Non è che far girare programmi in FORTRAN rendesse l'uso del 650 (o di qualsiasi altra macchina dell'epoca) molto più comodo.
- Se la macchina non aveva un lettore di nastro magnetico o un hard disk sufficientemente capiente (e non ce n'erano molti all'epoca), il compilatore FORTRAN era esso stesso codificato su schede, e occorreva far leggere al lettore di schede del 650 le schede del programma in FORTRAN insieme con le schede del compilatore. In altre parole, il programma FORTRAN era l'input del compilatore.
- Se tutto andava bene, il 650 produceva in output le schede perforate contenenti il programma da far girare tradotto in linguaggio macchina
- A questo punto le schede potevano di nuovo essere date in input al 650 per l'esecuzione. Il risultato veniva di nuovo fornito su schede perforate, che potevano essere stampate con opportuna stampante.¹⁷

Nota a margine: Donald Knuth e l'IBM 650

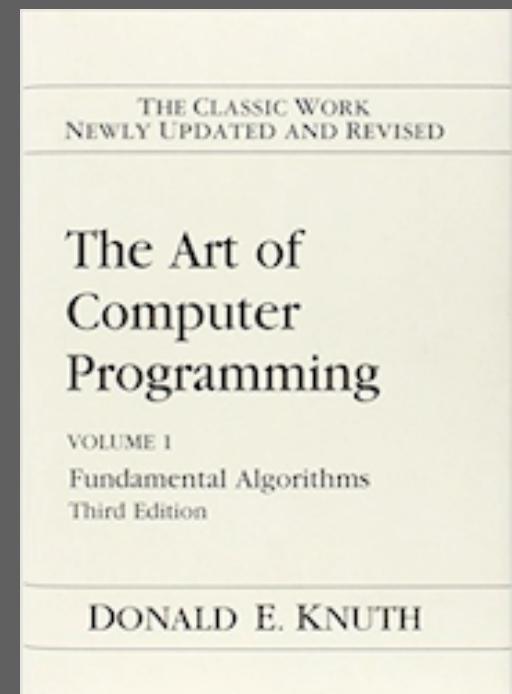
- **Donald Knuth** (1938) è uno dei giganti dell'informatica, ed è soprattutto noto per la sua opera principale, **The Art of Computer Programming** (a volte indicata come **TAOCP**)
- Il lavoro fu pubblicato in 4 volumi a partire dal 1968, ed è considerato non solo il primo, ma ancora il miglior testo completo sull'argomento. Possiamo tranquillamente affermare che qualsiasi altro testo di programmazione deriva dal TAOCP.
- Knuth completò la scrittura del manoscritto nel 1966, quando aveva solo 28 anni: circa 3000 pagine che diventavano 2000 stampate.
- Praticamente, era trattato qualsiasi fondamento teorico (come ricorsione e complessità) e qualsiasi algoritmo: ordinamento, ricerca, matematico, analisi sintattica, manipolazione di stringhe, parsificazione, generazione di numeri casuali, etc. etc. etc...

Nota a margine: Donald Knuth e l'IBM 650

- “The Art”, una vera e propria bibbia dell’arte di scrivere programmi, ha ricevuto innumerevoli riscontri positivi:
- La quarta di copertina della terza edizione del primo volume del TAOCP riportava una citazione di Bill Gates:

“If you think you're a really good programmer... read (Knuth's) Art of Computer Programming... You should definitely send me a résumé if you can read the whole thing”

- Il New York Times ha definito il TAOCP *“l’opera che ha definito una professione”*.
- La rivista American Scientist ha incluso il TAOCP nella lista dei *“100 or so Books that shaped a Century of Science”* (riferendosi al XX secolo)



Nota a margine: Donald Knuth e l'IBM 650

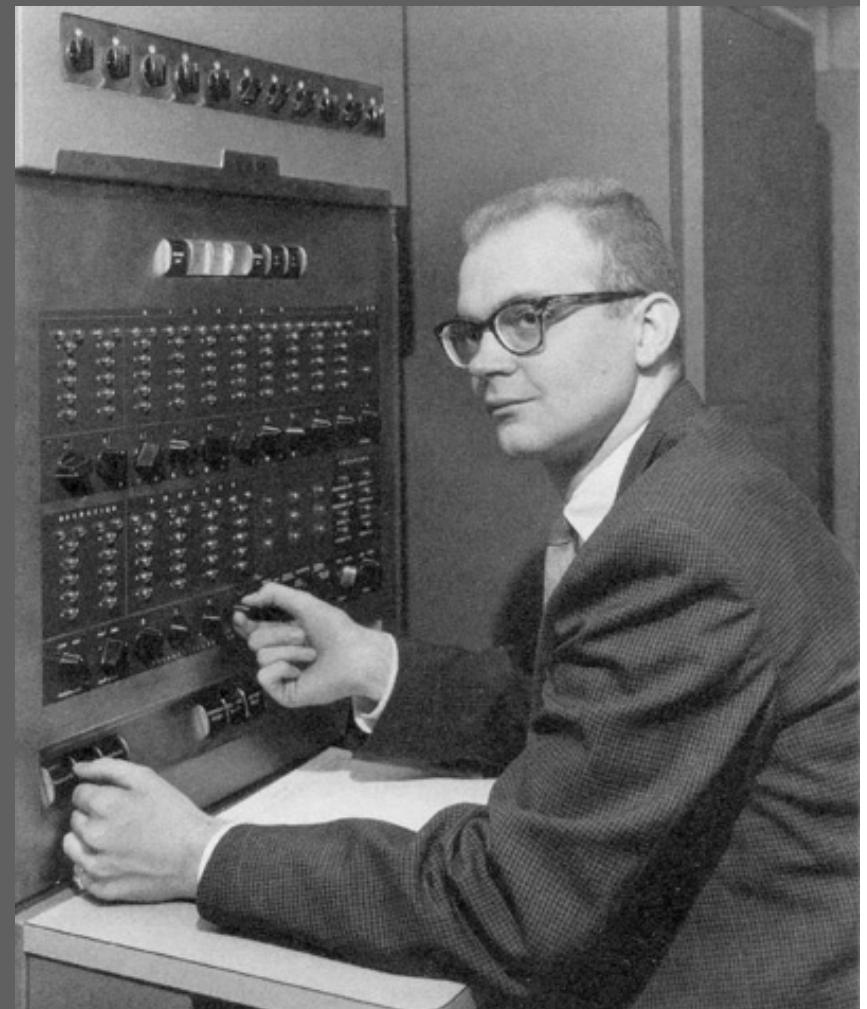
- Ma Donald Knuth non è noto solo per il TAOCP. Nel 1976, dovendo preparare la seconda edizione del secondo volume, constatò la necessità di un sistema di composizione di testi al computer. Otto anni dopo aveva sviluppato il **TeX**.
- Knuth ha fornito importanti contributi anche nel campo dell'informatica teorica e dei compilatori.
- A livello di attività pubblica si è distinto per essersi schierato contro la concessione dei brevetti software.
- Knuth incontra l'IBM 650 come studente, nella seconda metà degli anni 50. Dopo averne letto il manuale d'uso, decide di riscriverne il codice del compilatore e dell'assemblatore perché pensa di poterne sviluppare una versione migliore...

Nota a margine: Donald Knuth e l'IBM 650

- È certo che la conoscenza acquisita da Knuth sugli algoritmi sia stata sviluppata sull'IBM 650, su cui sono stati tra l'altro testati i tantissimi esercizi che accompagnano il TAOCP.
- Non a caso infatti, la dedica sui volumi del TAOCP recita:

This series of books is affectionately dedicated to the Type 650 computer once installed at Case Institute of Technology, with whom I have spent many pleasant evenings

- Nella foto, del 1958, vediamo Knuth ventenne alla console del 650



Nota di colore: bit e bytes

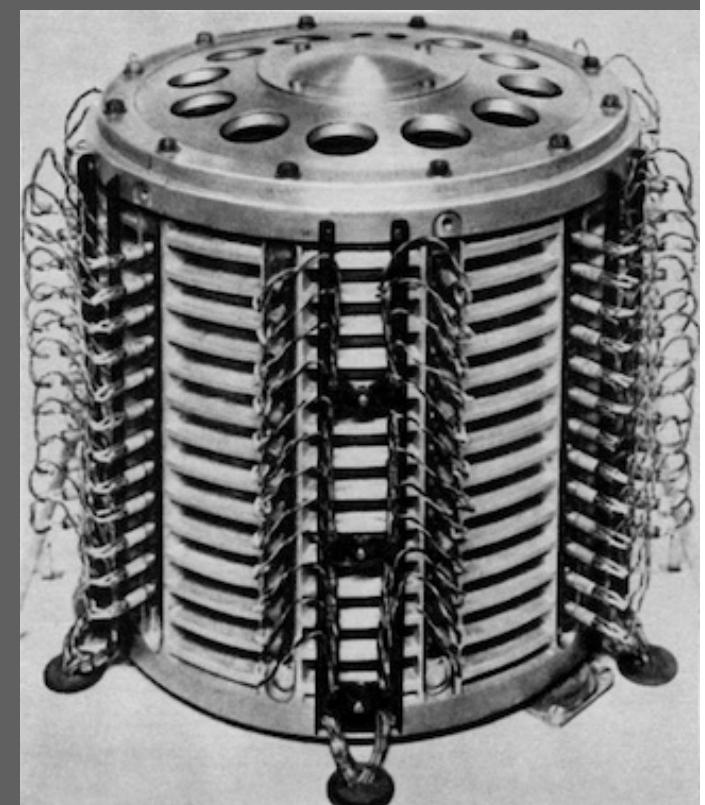
- Abbiamo già usato la parola **bit**: in inglese *bit* significa “*pezzetto di qualcosa*” e già nel 1936 Vannevar Bush scriveva di “*bits of information*” memorizzati su schede perforate.
- Nel 1948 *Claude Shannon* userà la parola “bit” nel suo fondamentale: “**A Mathematical Theory of Communication**”, ma il termine gli sarebbe stato suggerito dal collega **John Tukey**, che in un articolo del 1947 aveva parlato di **Binary digit**, o **bit**.
- Nel 1956 Werner Buchholz (lavorando al progetto dell’IBM 7030) inventa la parola **byte**: una mutazione della parola *bite* (facile a confondersi con *bit*) che in inglese significa “*morsa, boccone*”.
- E ci volle un po’ di tempo perché la dimensione di un byte si stabilizzasse a 8 bit: un byte poteva anche essere lungo 4 (per codificare un numero decimale) o 6 (un carattere alfanumerico) ⁵² bit.

Memorie a tamburo rotante

- Come abbiamo visto, l'IBM 650 nella configurazione di base era dotato di una sola memoria principale: era una **Memoria a Tamburo Rotante**, o **Magnetic Drum Memory**. (**MDM**).
- Questa tecnologia, in effetti già disponibile dagli anni '30, fu negli anni '50 migliorata in modo da fornire spazi di memorizzazione (qualche decina di Kword) e tempi di accesso migliori delle MDL.
- Con l'invenzione delle memorie a nucleo magnetico, più veloci ma più costose, e con lo sviluppo dei primi sistemi multiprogrammati, si incominciarono a costruire computer dotati di due livelli di memoria primaria: Memorie a nucleo magnetico + memorie a tamburo rotante.
- In effetti sia le MDM che le MCM erano memorie non volatili, ma troppo piccole per poter essere usate come memoria secondaria, per la quale si continuaron ad usare nastri magnetici e hard disk.

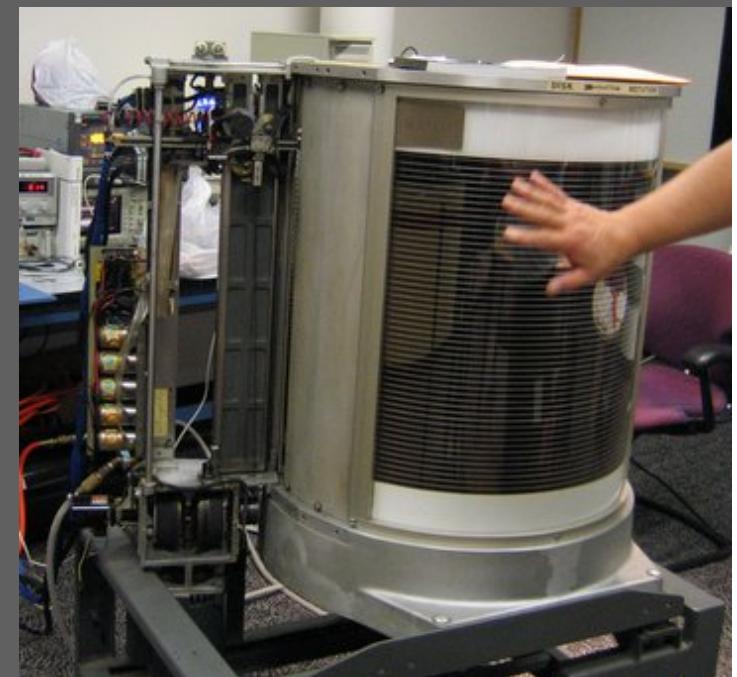
Memorie a tamburo rotante

- Inventate nel 1932 da **Gustav Tauschek**, le MDM erano costituite da un tamburo rotante la cui superficie verticale era ricoperta da bande di materiale ferromagnetico (un po' come negli hard disk).
- Più testine di lettura/scrittura venivano disposte attorno al tamburo in una o più file verticali, in corrispondenza delle bande magnetiche.
- I programmatori posizionavano le istruzioni in memoria in modo da ridurre il tempo necessario alla word contenente la prossima istruzione da eseguire di ruotare così da posizionarsi sotto la testina di lettura.
- Ma con l'inizio degli anni '60 le MDM furono velocemente rimpiazzate dalla tecnologia superiore delle memorie a nucleo magnetico, e dagli hard disk.



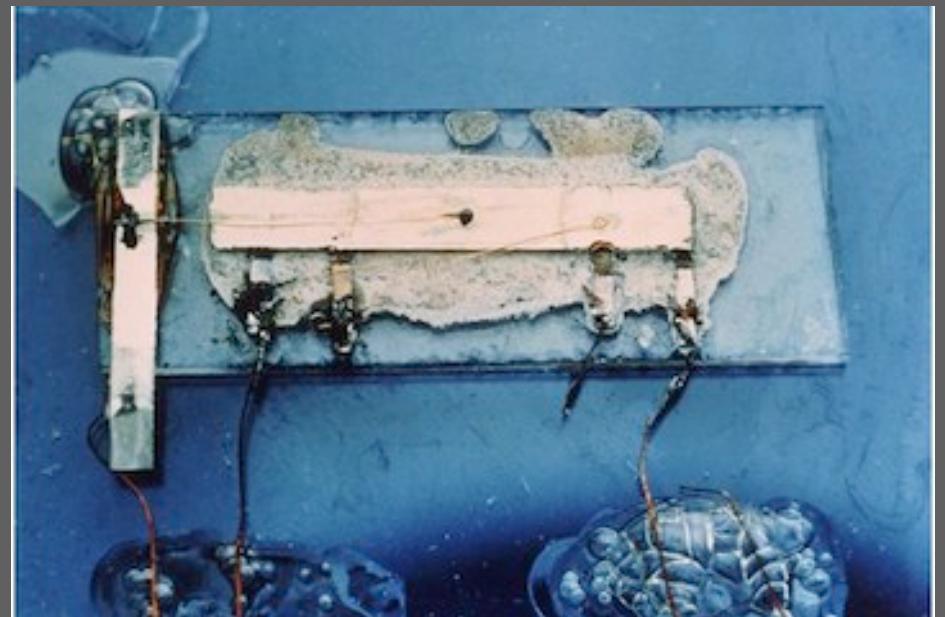
Finalmente, il primo hard disk

- Con il computer IBM RAMAC 305 per applicazioni commerciali, presentato nel 1956, fa finalmente la comparsa il primo hard disk, concettualmente molto simile a quelli moderni, ovviamente con dimensioni, capacità e velocità assai diverse.
- Il primo hard disk della storia aveva una capacità di circa 5 Megabit, memorizzati su un totale di 50 piatti magnetici di 61 cm di diametro.
- Ogni piatto aveva materiale magnetico su entrambi i lati, e 50 bracci meccanici su cui erano montate 2 testine (per la superficie inferiore e superiore) che si muovevano indipendentemente tra un piatto e l'altro.
- I piatti ruotavano a 1200 RPM trasferendo circa 60 Kbit di dati al secondo



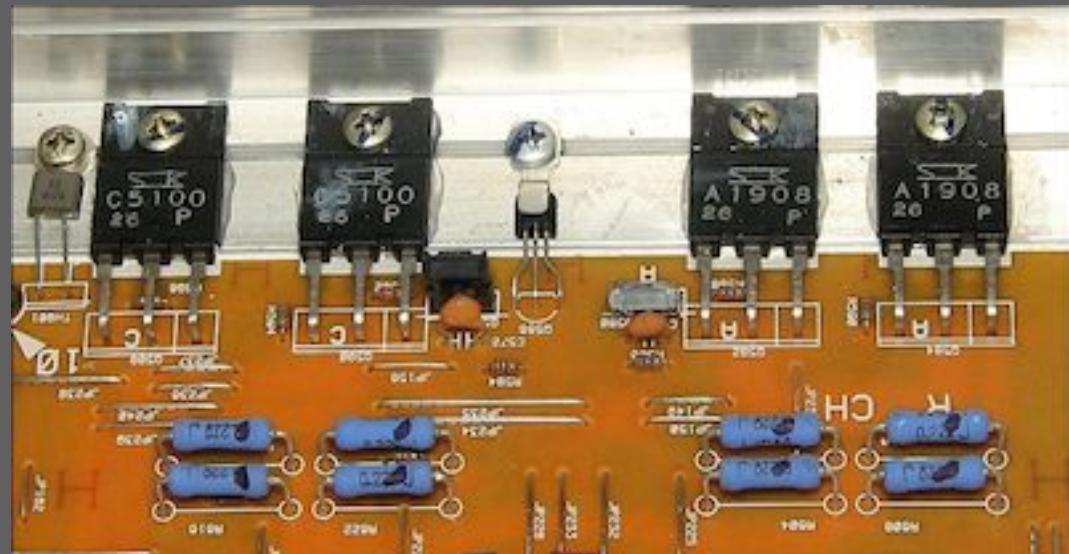
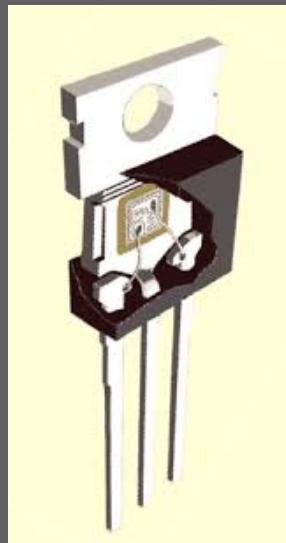
E finalmente, il primo circuito integrato

- Gli anni ‘50 si chiudono con un avanzamento tecnologico fondamentale per tutta la storia dell’informatica (e per il progresso tecnologico in generale): la realizzazione del primo **circuito integrato (IC)** da parte di **Jack Kilby** della **Texas Instruments**, nel 1958. Per questa invenzione Kilby riceve il premio Nobel nel 2000.
- In realtà, idee simili erano già state sviluppate in precedenza, ma mai portate a compimento, da W. Jacobi nel 1949 e da W. Dummer nel 1952. Poco dopo Kilby, **Robert Noyce** realizzò indipendentemente un circuito integrato.
- Il primo IC conteneva solo un flip flop, ossia pochi transistor, ma Kilby aveva dimostrato che questi componenti discreti si potevano **integrare e miniaturizzare** su un unico strato di semiconduttore.



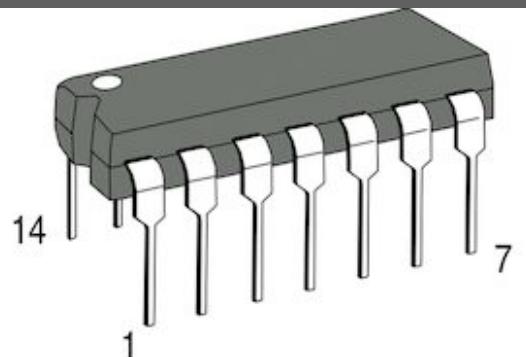
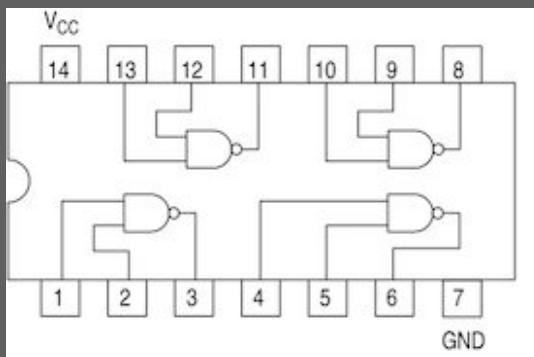
E finalmente, il primo circuito integrato

- Infatti, è vero che l'invenzione del transistor aveva reso obsolete le valvole termoioniche, ingombranti e inaffidabili.
- Ma anche i transistor, per quanto più piccoli, erano oggetti che occupavano uno spazio relativamente grande, dovuto non tanto alla fettina di materiale semiconduttore che realizzava il transistor stesso, ma alla capsula di plastica che lo conteneva, e ai 3 piedini elettrici che permettevano di collegarlo agli altri componenti di un circuito elettronico.



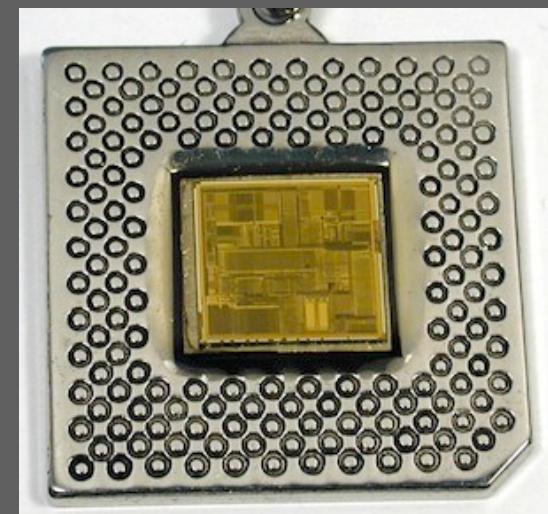
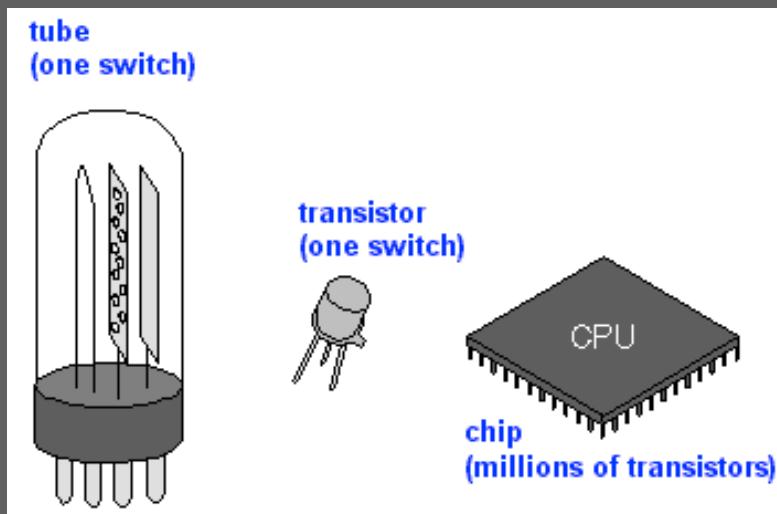
E finalmente, il primo circuito integrato

- Un IC invece permetteva di mettere assieme più transistor in uno spazio veramente piccolo, collegandoli fra di loro a seconda del tipo di circuito logico che si voleva costruire.
- La fettina di silicio contenente i transistor veniva poi inserita in un unico contenitore di materiale plastico dotato di un numero adeguato di piedini elettrici per collegare l'IC ad altri componenti.
- In questo esempio, l'IC contiene 4 porte NAND, per un totale di 24 transistor (se si usano solo transistor, ce ne vogliono 6 per realizzare una porta NAND)



E finalmente, il primo circuito integrato

- I primi IC contenevano ben pochi transistor, ma col tempo si è riusciti ad inserirne sempre di più su un'unica fettina di silicio, fino a realizzare, allo stato attuale, processori multi core formati da qualche miliardo di transistor che occupano una superficie di 400÷600 mm².
- La **Legge di Moore** stima che il numero di transistor che possiamo mettere in un IC raddoppia ogni 18/24 mesi. La “legge”, formulata addirittura nel 1965 da **Gordon Moore**, (cofondatore di **Intel** con **Robert Noyce**), fin’ora si è rivelata abbastanza precisa, ma nessuno può dire per quanto resterà valida. Insomma:



Gli anni ‘60: la nascita dei computer moderni

- Negli anni ’60 verranno concepite, sviluppate e sperimentate molte soluzioni architetturali volte ad aumentare la velocità di esecuzione dei programmi, che diventeranno parte del funzionamento dei moderni computer.
- In molti casi, queste innovazioni troveranno piena attuazione solo alcuni anni più tardi, quando la tecnologia permetterà di realizzarle in modo affidabile, efficiente ed economico.
- E ci vorrà qualche anno perché i circuiti integrati, appena inventati, diventino parte dell’architettura dei computer.
- Gli anni ‘60 vedono anche la produzione di alcuni modelli di computer entrati nella storia dell’informatica per varie ragioni.

Il PDP-1: Programmed Data Processor-1

- Il 1960 si apre con la commercializzazione, da parte della **Digital Equipment Corp.** (più nota come **DEC**) del primo di una fortunata serie di computer, che arriverà fino al PDP-16 nei primi anni '70.
- Il PDP-1, dal costo base di circa 120.000 \$ del tempo, e venduto in 53 esemplari fino al 1969, fu progettato esplicitamente per fornire un'interfaccia “user-friendly”. È importante per la storia dell'informatica perché detiene vari primati:
- Primo computer commerciale ad usare un monitor
- Primo computer ad ospitare un videogame “moderno”: spacewar
- Primo computer su cui sia stato sviluppato un text editor, un word processor, un debugger interattivo. E tra i primi su cui sia stato sviluppato un programma per giocare a scacchi e programmi di computer music.

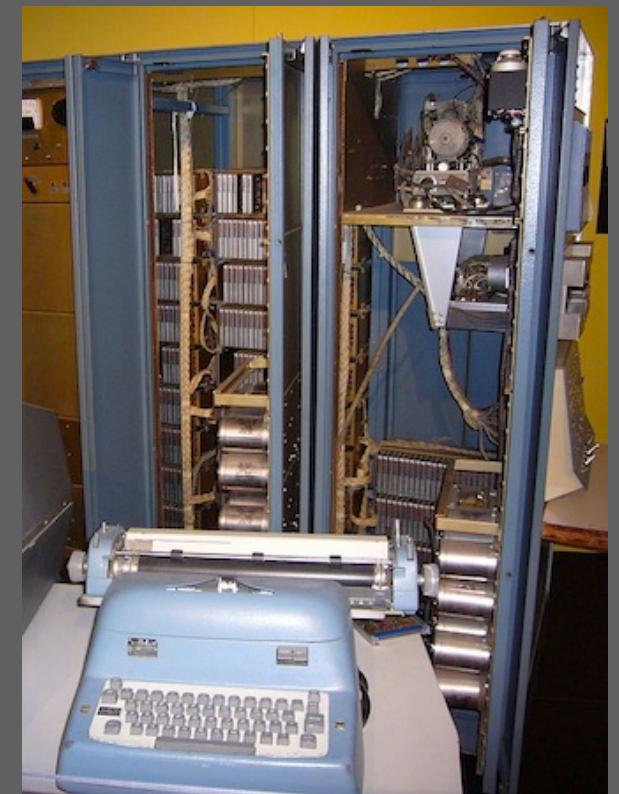
Il PDP-1: Programmed Data Processor-1

- Più in generale fu sul PDP-1 che incominciò a formarsi la cultura **hacker** nel campo dei computer: persone che si divertivano a studiare nuove possibilità e potenzialità scrivendo programmi originali e/o modificando l'hardware esistente. Un campo che in quel periodo era, ovviamente, quasi completamente inesplorato.
- Usava un lettore di nastro perforato come memoria di massa principale, e 8 Kbyte di magnetic core memory.
- Eseguiva circa 100.000 istruzioni al secondo, con un tempo di accesso alla core memory di 5 μsecondi.
- L'output era riversato su nastro perforato o sul monitor.



Il PDP-1: Programmed Data Processor-1

- Il PDP-1 era completamente costruito con transistor e diodi (in tutto quasi 6.000) montati su schede (chiamate system building blocks) a loro volta montate all'interno di un rack verticale.
- Questo soluzione modulare permetteva una flessibilità di configurazione piuttosto innovativa.
- La presenza di una console telescrivente e di un monitor rendeva il lavoro al PDP-1 molto simile a quello dei giorni nostri
- Il programmatore poteva sedersi alla console, scrivere un programma con l'editor di testi e farlo subito girare, guardando il risultato sulla telescrivente stessa o sul monitor.

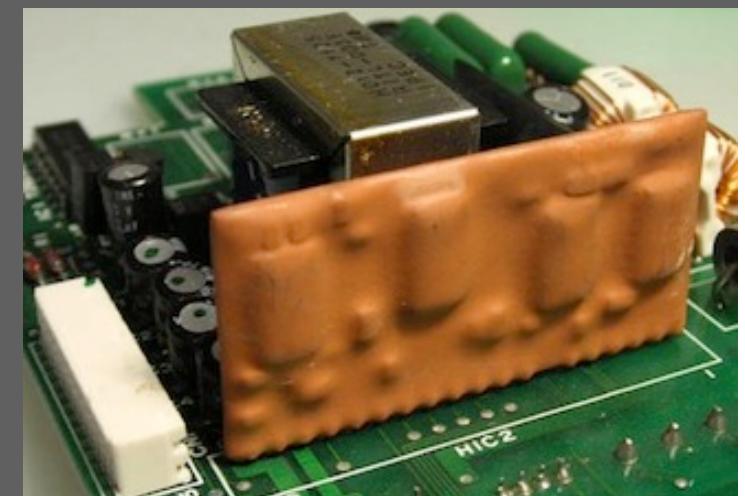


Varie innovazioni tecniche

- Nella prima metà degli anni '60 vengono progettati e prodotti diversi computer che contengono innovazioni tecniche poi adottate stabilmente da tutti i computer degli anni successivi.
- Nell'**ILLIAC II**, operativo dal 1962, compare l'idea di **pipeline**, suddivisa in tre fasi: *fetch*, *decode*, *execute*. (in realtà, un concetto simile era già presente nello Z1 del 1939)
- Nel **IBM 7030** (noto anche come “**stretch**”), anch’esso del 1962, oltre alla pipeline vengono implementate la **protezione della memoria**, i **byte da 8 bit** (ma poteva usare anche byte da 4 o 6 bit), e la **memoria interlacciata**.
- E già nel 1961 nell'**ATLAS**, al tempo il computer più potente in funzione, erano state realizzate la **paginazione** e la **memoria virtuale**, e il **B5000** aveva già implementato la **segmentazione**
(ci torneremo nella parte sui sistemi operativi)

La terza generazione di computer

- La serie **IBM/360**, a partire dal 1964, inaugura la **terza generazione** di computer, basata cioè sull'uso di circuiti integrati (IC).
- Retrospettivamente è diventata abitudine indicare come la **seconda generazione** di computer quelli costruiti a transistor, ossia più o meno dal 1955 al 1963, e come **prima generazione** quelli costruiti a valvole termoioniche.
- Ma i confini temporali e tecnologici non sono precisi. Il primo computer della serie 360 non usava veri e propri circuiti integrati, ma cosiddetti **circuiti ibridi**, un passo intermedio tra transistors e IC veri e propri, composti da un unico blocco contenente semiconduttori e componenti elettrici (resistenze, condensatori).
- I primi computer ad usare IC furono i Burroughs B2500 e B3500 nel 1968.



La serie IBM/360

- Questa fortunata serie di mainframe, prodotta dal 1964 al 1978, fu la prima a mettere a disposizione un'ampia gamma di computer, adatti ad applicazioni diverse e a diverse capacità di spesa.
- Il modello di base della serie aveva una velocità di esecuzione variabile da 0,0018 a 0,034 **MIPS** (**M**illions of **I**nstructions **P**er **S**econd), e il più veloce era circa 50 volte più veloce.
- I vari modelli avevano una memoria primaria da un minimo di 8 KByte fino a 1 MByte di MCM (in realtà si poteva arrivare fino a 8 MegaByte)
- La memoria secondaria era su nastro magnetico o su hard disk rimovibile con una capacità massima di 7,2 MByte



La serie IBM/360

- La serie 360 adotta una serie di soluzioni tecniche molto interessanti dal punto di vista storico, perché contribuiranno in maniera determinante a definire gli standard degli anni successivi.
- **L'adozione di byte di dimensione fissa da 8 bit.** Nonostante ci fosse stata una forte pressione da parte dei vertici finanziari a usare byte da 4 o 6 bit (come nel 7030), che permettevano di risparmiare spazio per quei dati che non avevano bisogno di 8 bit per essere rappresentati.
- **Memoria principale indirizzabile a byte** (e non più a singoli bit o ad intere word)
- **Word di dati da 32 bit.**
- **L'uso esteso del microcodice in tutti i processori**, che permetteva, se necessario, di cambiare l'instruction set di una macchina cambiando il dispositivo su cui era memorizzato il microcodice.¹⁶⁷

La serie IBM/360

- **16 registri general purpose da 32 bit ciascuno**
- **Una Program Status Word da 64 bit**
- **Un meccanismo di interrupt mascherabili e gerarchici**
- **L'uso di un TLB per la traduzione degli indirizzi.**
- **Istruzioni di lunghezza variabile: 2, 4 o 6 byte.** Come vedremo quando parleremo della rivoluzione RISC degli anni '80, ciò si rivelerà un limite alla velocità di esecuzione dei programmi.
- **Istruzioni macchina complesse con 2 operandi in memoria.** Anche questo tipo di istruzioni sarà oggetto di critica e ripensamenti, che porteranno allo sviluppo delle architetture RISC.
- **L'introduzione del Direct Memory Access (DMA),** un canale diretto di comunicazione tra le memorie primaria e secondaria. ¹⁶⁸

L’Olivetti Programma 101 aka “Perottina”

- Sviluppata da **Pier Giorgio Perotto** tra il 1962 e il 1964 alla **Olivetti** di Ivrea, viene presentata alla fiera di New York del 1964, e vendette circa 40.000 esemplari, circa il 90% dei quali negli Stati Uniti.
- Sebbene fosse presentata come calcolatrice, era di fatto un piccolo computer, in quanto dotata di 10 registri da 22 bit e programmabile in un linguaggio macchina dotato di istruzioni di salto condizionato.
- Era costruita con transistors, e i registri erano costituiti da piccole delay lines. Una scheda di plastica dotata di banda magnetica conteneva il programma da eseguire, quindi facilmente trasportabile da una macchina ad un’altra, un’idea rivoluzionaria per i tempi.
- Grazie al suo design futuristico, la P101 è esposta al Museum of Modern Art di New York.



Seymour Cray e il CDC 6600

- Seymour Cray è una figura leggendaria nel campo dei supercomputer, e la sua fama inizia con la progettazione del **CDC 6600**, commercializzato a partire dal 1965 dalla Control Data Corporation.
- Il CDC 6600 è generalmente considerato il primo **supercomputer** di successo della storia, ed è rimasto il computer più veloce della sua generazione fino al 1969, quando fu superato dal CDC 7600.
- Il CDC raggiungeva una potenza di calcolo di circa 1 milione di istruzioni floating point al secondo, ossia: **1 MegaFLOPS**. Nel 1969, il 7600 raggiungerà il record di 40 MFLOPS.
- L'unità di misura in FLOPS è ormai entrata nell'uso comune per indicare le prestazioni dei computer, e in particolare dei supercomputer, in sostituzione della più generica MIPS

Seymour Cray e il CDC 6600

- La serie 6000, e il 6600 in particolare, portarono un certo numero di idee innovative, a partire dall'uso di transistor al silicio, recentemente introdotti in commercio e molto più veloci di quelli al germanio.
- Ma una delle intuizioni fondamentali di Cray, poi consolidate in tutti i processori del futuro, fu di separare nettamente le operazioni di esecuzione delle istruzioni da altre operazioni necessarie a far funzionare un computer nel suo complesso (principalmente l'I/O).
- In un normale computer moderno, la CPU è assistita da un certo numero di altri circuiti integrati, che permettono al processore di comunicare con i dispositivi veloci, come la RAM e le schede video (**northbridge**) e con gli altri dispositivi meno veloci (**southbridge**)
- Almeno concettualmente, questi circuiti integrati, noti nel loro insieme come **chipset**, sono separati, e sono collegati insieme alla CPU sulla motherboard del computer

Seymour Cray e il CDC 6600

- Nei primi anni ‘60 invece, tutte le funzioni di comunicazione con le varie periferiche, memoria principale inclusa, venivano inglobate nella CPU, la quale doveva ovviamente anche eseguire le istruzioni dei programmi.
- Il risultato erano CPU molto complesse, e quindi molto grandi, e in cui si dilatavano i tempi di trasferimento del segnale da un modulo all’altro del processore.
- Come conseguenza, il ciclo di clock della macchina doveva essere lungo abbastanza da permettere il trasferimento dei segnali da un punto all’altro del processore.
- In più, negli anni ‘60 le CPU erano più lente della RAM: bastavano 2 cicli di clock per leggere un dato/istruzione in RAM, ma ce ne volevano 15 per una moltiplicazione. Era questa differenza che Cray pensò di sfruttare.

Seymour Cray e il CDC 6600

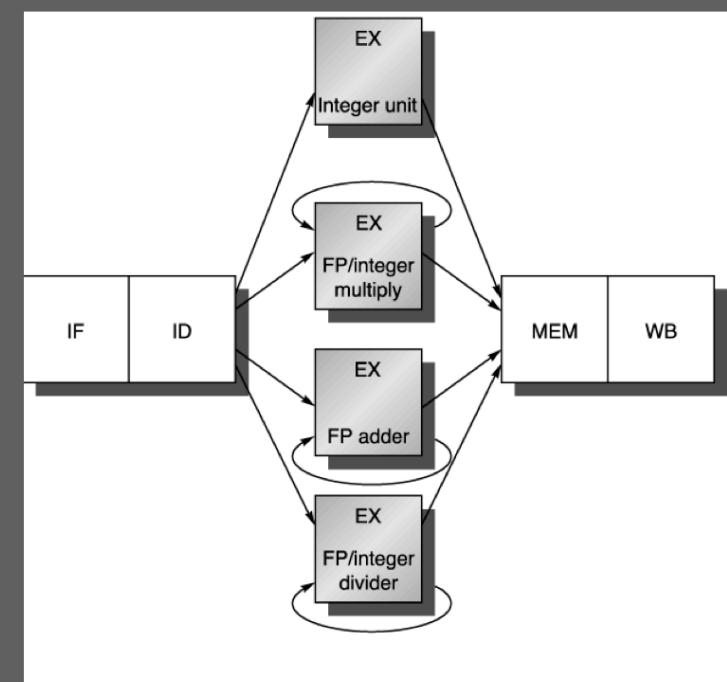
- La CPU del 6600 eseguiva solo semplici istruzioni aritmetiche e logiche, ed era quindi anche molto più semplice e piccola.
- Sfruttando anche i nuovi transistor al silicio, poteva funzionare con un ciclo di clock di 100 ns, o 10 MHz, dieci volte più veloce dell'IBM 7030, fino a quel momento considerato la macchina più veloce sul mercato.
- Un instruction set semplice faceva del 6600 l'antenato delle architetture RISC degli anni '80. Una idea fondamentale quindi, tuttavia ignorata per circa 15 anni (vedremo meglio più avanti)
- Se la CPU non doveva occuparsi di gestire i dispositivi di I/O e le comunicazioni con la RAM direttamente, poteva operare in parallelo ad esse, migliorando le prestazioni della macchina.
- Ma bisognava ancora occuparsi di I/O e RAM

Seymour Cray e il CDC 6600

- Invece di utilizzare un processore periferico che si occupasse di ciascun dispositivo, Cray pensò di usare un altro processore prodotto dalla CDC, usato come unico **Processore Periferico (PP)**.
- Il PP si occupava delle comunicazioni tra la CPU del 6600 e le varie periferiche eseguendo in Round Robin le istruzioni necessarie a comunicare con ciascuna periferica.
- In altre parole, il PP funzionava come una **CPU multithreaded a grana fine** (in grado di gestire fino a 10 thread contemporaneamente)
- Per rimarcare la separazione tra operazioni aritmetico-logiche e operazioni che coinvolgono l'accesso in RAM, il 6600 aveva solo due istruzioni per accedere alla memoria: read e write. (un altro principio fondamentale delle architetture RISC)

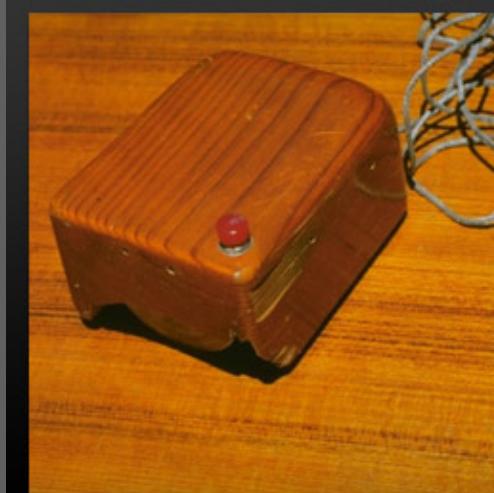
Seymour Cray e il CDC 6600

- Il 6600 fu il primo esempio di computer con una CPU ad architettura che oggi chiameremmo **superscalare**.
- Il processore era infatti dotato di 10 unità funzionali che potevano eseguire le istruzioni in parallelo. Era il primo esempio di sfruttamento del parallelismo all'interno delle istruzioni di un programma, più noto come **Instruction Level Parallelism (ILP)** e oggi usato da tutti i processori moderni.
- Tutto funzionava perché prelievo e decodifica delle istruzioni richiedevano molto meno tempo della loro esecuzione, cosicché nuove istruzioni potevano essere avviate all'esecuzione senza attendere il completamento di quelle precedenti purché fossero indipendenti: il pipelining.



Nota di colore: l'invenzione del mouse

- Nel 1964 **Douglas Engelbart** concepisce e realizza, insieme all'ingegnere Bill English, un dispositivo in grado di indicare le coordinate di una posizione su uno schermo: il **mouse**.
- L'idea in realtà era parte di una visione più ampia delle potenzialità dei computer di espandere l'espressività degli esseri umani (un po' come il linguaggio che usiamo definisce le cose che possiamo pensare e su cui possiamo ragionare)
- Ispirato dalle idee di Vannevar Bush, Engelbart concepirà e svilupperà negli anni '60 molte altre idee fondamentali per la human-computer interaction: gli schermi bitmapped, le interfacce grafiche e gli ipertesti.



I minicomputer

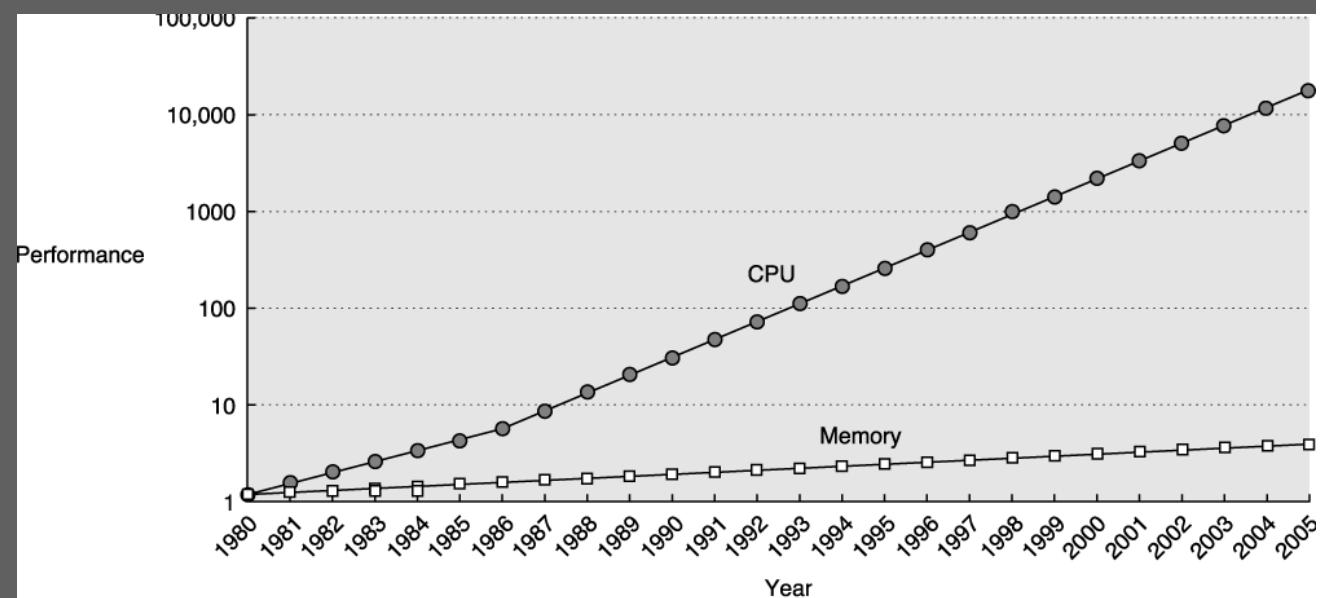
- Con il lancio del PDP-8 nel 1965 nasce una nuova categoria di macchine, i **minicomputer**, un termine però vago e ormai obsoleto.
- In generale si riferiva a calcolatori più economici (fino a 250.000 \$ circa), più piccoli (delle dimensioni di un armadio anziché di una stanza) e con meno periferiche dei mainframe.
- Costituivano però una nuova fetta di mercato e più di 100 compagnie si lanciarono nella loro produzione.
- Incominciarono a declinare negli anni ‘80, con la diffusione di microprocessori economici (la serie x86) che permisero di sostituire i minicomputer e i loro terminali video con workstation, PC e file server collegati da una rete locale.



La comparsa delle prime cache

- Negli anni ‘60, i tempi di accesso ad una informazione in RAM erano solo leggermente superiori al tempo necessario a leggere un registro.
- Ma la crescita della frequenza del clock delle nuove macchine suggeriva un trend che avrebbe reso la RAM sempre più lenta del processore che la doveva usare, creando così un collo di bottiglia.
- Andamento dei rapporti di prestazione tra CPU e RAM negli anni, posto a 1 il rapporto di prestazioni nel 1980. Miglioramento RAM: circa 7% l’anno.

Miglioramento CPU:
35% l’anno fino al
1986, 55% negli
anni successivi.



La comparsa delle prime cache

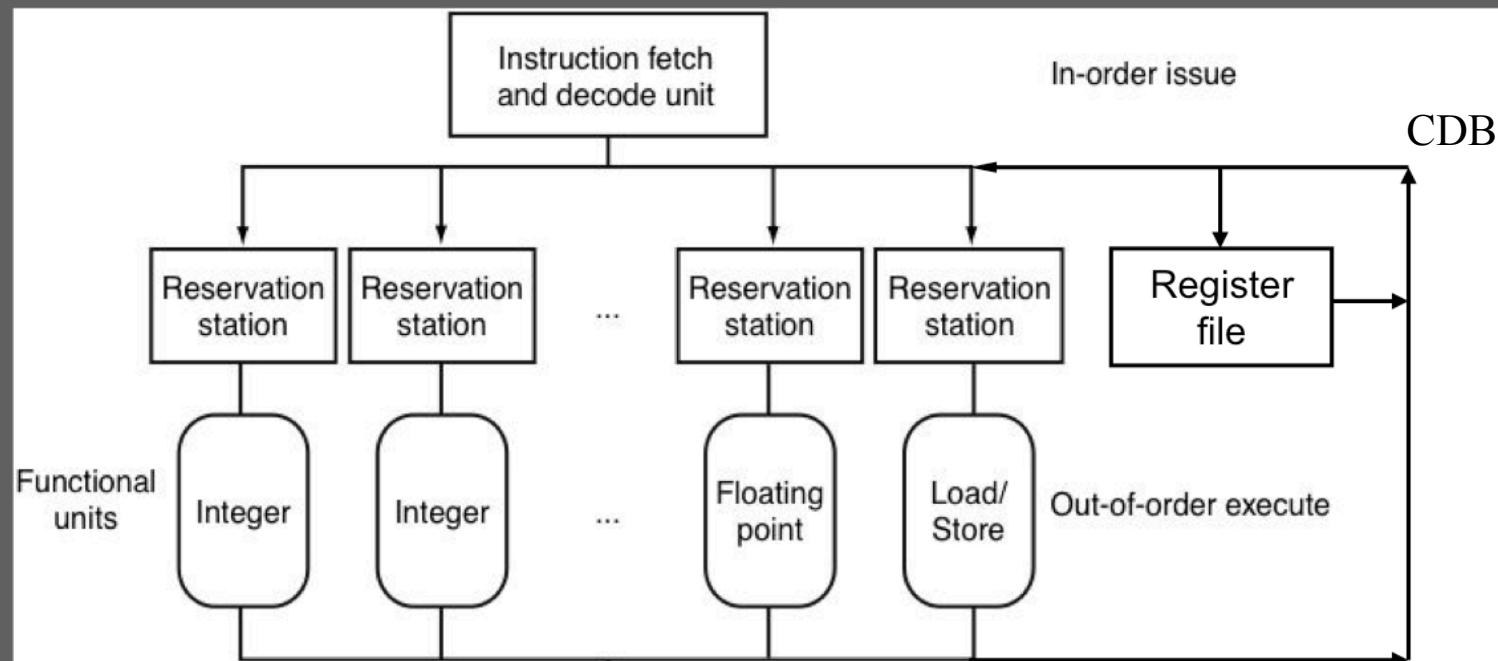
- Nel 1965 Maurice Wilkes pubblica un articolo dove descrive l'idea di utilizzare uno strato di memoria intermedio fra la RAM e la CPU, che sia più veloce (e quindi più costoso) e che possa contenere i dati e le istruzioni di più immediato utilizzo. La **cache**, appunto.
- Il primo computer commerciale a contenere una cache è l'IBM 360/85, nel 1968.
- Lo stesso anno, un esperimento condotto usando 11 diversi programmi fatti girare sul 360/85 (con cache) e sul ben più veloce 360/91 (senza cache) mostra l'importanza della cache: 8 degli 11 programmi giravano più velocemente sul 360/85
- Al giorno d'oggi, tutti i processori adottano (più livelli di) cache

Lo schema di Tomasulo

- Nel 1967 **Robert Tomasulo**, della IBM, inventa uno schema (a volte chiamato “*algoritmo di Tomasulo*”) che permette alla CPU di eseguire le istruzioni di un programma **out-of-order**, ossia in un ordine diverso da quello con cui vengono prelevate dalla RAM.
- In un normale datapath, se una istruzione A non può essere eseguita perché manca uno dei suoi operandi, le istruzioni che vengono dopo A devono anch’esse aspettare, anche se sono indipendenti da A.
- Lo schema di Tomasulo modifica radicalmente il datapath del processore, in modo tale da permettere a istruzioni indipendenti fra loro di essere eseguite non appena i loro operandi sono disponibili.
- Questa soluzione rende il processore più complesso, ma ne aumenta di molto le prestazioni, ed è ormai adottata da tutti i processori moderni.¹⁸⁰

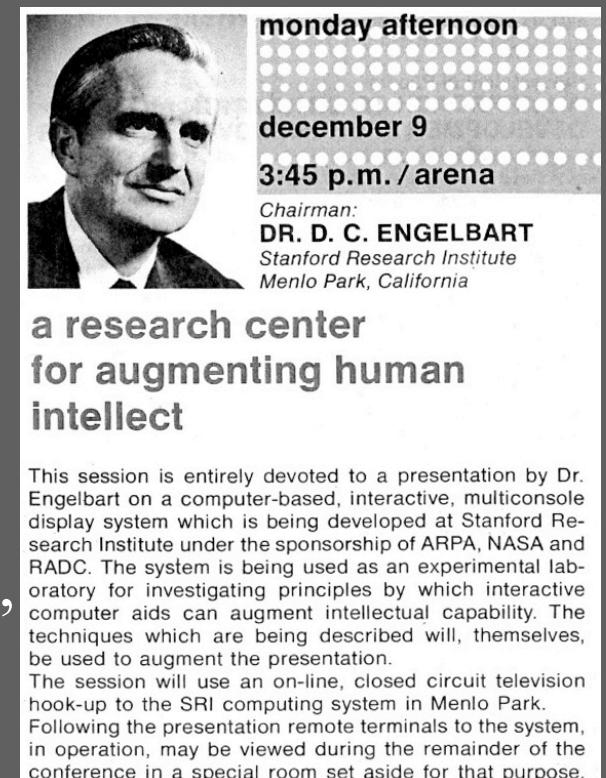
Lo schema di Tomasulo

- Lo schema di Tomasulo si basa sulle **Reservation Station** (RS) complesse memorie associative in cui sostano le istruzioni che devono essere eseguite, in attesa degli operandi di cui hanno bisogno.
- Non appena un operando è disponibile, viene smistato mediante il **Common Data Bus** (CDB) verso tutte le RS in cui c'è una istruzione che ne ha bisogno per essere eseguita. Le istruzioni non devono più essere eseguite nell'ordine con cui sono state prelevate dalla RAM



Nota di colore: the mother of all demos

- Il 9 dicembre 1968 **Douglas Engelbart** presenta alla Fall Joint Computer Conference a San Francisco la dimostrazione del funzionamento di un computer denominato **oN-Line System** (o **NLS**)
- Nella demo vengono presentate tutte le caratteristiche fondamentali di un moderno computer come lo intendiamo oggi: **finestre, ipertesti, grafica** e navigazione efficiente, l'uso del **mouse** e **word processing**.
- Quella che sarà retrospettivamente chiamata **“the mother of all Demo”** ha una influenza fondamentale sullo sviluppo dei computer del futuro negli anni ‘70 (a Xerox Parc), ‘80 (dalla Apple) e ‘90 (Microsoft).
- Engelbart, considerato fino a quel momento una figura stravagante, dopo 90 minuti era diventato, per tutta la comunità scientifica, un genio.



Gli anni ‘70: il PDP-11

- Gli anni ‘70 si aprono con la commercializzazione del **PDP-11**, forse il computer (al tempo era considerato un minicomputer) più famoso e di successo della **Digital Equipment Corporation**, o **DEC**.
- Dei vari modelli, prodotti fino all'inizio degli anni ‘90, ne furono venduti in tutto circa 600.000 esemplari.
- La sua architettura ebbe una forte influenza su futuri processori e influenzò anche il progetto di futuri sistemi operativi. Su un PDP-11 girò la prima versione ufficiale del sistema operativo Unix.
- Tra l'altro, fu progettato esplicitamente per la produzione di massa, ossia i suoi componenti potevano essere facilmente e velocemente assemblati da addetti non specializzati.

Il PDP-11

- Il PDP-11 era un computer di prestazioni medie, aveva solo 8 registri generali, una memoria primaria di 64 Kbyte e costava poche decine di migliaia di dollari, un prezzo quindi accessibile a molti.
- Una caratteristica fondamentale del PDP-11 era la cosiddetta **ortogonalità dell'Instruction set**: qualsiasi modalità di indirizzamento dei dati poteva essere utilizzata da qualsiasi istruzione. Una flessibilità che rendeva molto comodo il lavoro dei programmatori e dei compilatori (rendeva anche lenta l'esecuzione dei programmi...).



Nota di colore: il floppy disk

- I **floppy disk** (sia il supporto magnetico che i dispositivi per usarli) incominciarono ad essere disponibili in commercio dal 1971.
- Erano dei semplici supporti di memoria magnetica permanente trasportabili, e quindi rendevano relativamente comodo il trasporto e lo scambio di dati e programmi, e il backup.
- I primi floppy avevano 8 pollici di diametro (20 cm) ridotti poi a $5\frac{1}{4}$ e $3\frac{1}{2}$ pollici, con capacità che andava da 360 a 1440 Kbyte.
- Negli anni '80 e '90 il loro uso era enormemente diffuso, ogni computer aveva un lettore di floppy e nei primi PC erano l'unico supporto di memoria secondaria.
- Iniziarono a declinare negli anni '2000, sostituiti dai CD e dalle memorie a stato solido.



La vera storia della Intel

“i buoni artisti copiano, i grandi artisti rubano”

Steve Jobs, 1984

(In realtà parafrasava Pablo Picasso, che aveva detto:

“i cattivi artisti copiano, quelli buoni rubano”)

- Alla Intel c'erano sicuramente dei grandi artisti... Del resto, come vedremo fra poco, **Federico Faggin** responsabile dello sviluppo del primo microprocessore della storia, il 4004, usava firmare i chip...

La nascita della Intel

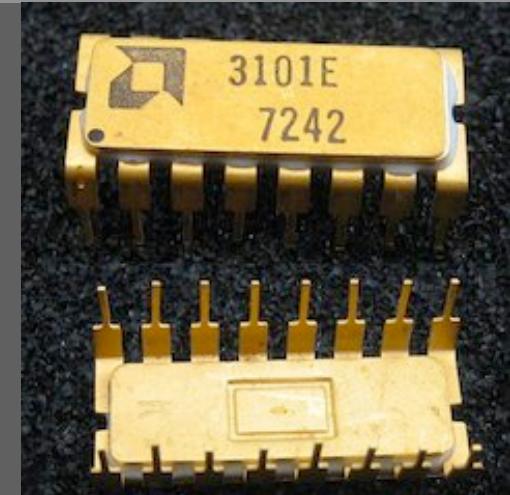
- Nel 1968 **Robert Noyce** e **Gordon Moore** lasciano la Fairchild Semiconductor e, il 18 luglio, fondano la **NM Electronics**, che un anno dopo diviene **Integrated Electronic Corporation**, poi abbreviata in **Intel**.
- Poco dopo assumono **Andrew Grove**, anch'egli dalla Fairchild, che dirigerà poi l'azienda negli anni '80 e '90, facendola diventare il più importante produttore di circuiti integrati e processori del mondo.
- La Intel nasce con l'intento di sviluppare e produrre memorie RAM a semiconduttori, e questa rimane la sua principale attività fino al 1980.
- Noyce e Moore infatti pensavano che le memorie a semiconduttore avrebbero presto rimpiazzato le MCM come memoria primaria dei computer. E i computer avrebbero avuto bisogno di sempre più chip di memoria, mentre la CPU sempre una sarebbe rimasta... 187

La nascita della Intel

- In effetti, già nel 1961 Bob Norman della Fairchild aveva proposto di memorizzare bit di dati usando flip flop a semiconduttori incapsulati in un unico circuito integrato.
- Nell'ottobre del 1961 La Texas Instruments consegna alla Air Force un piccolo computer in cui poche centinaia di bit sono memorizzate usando semiconduttori.
- Nel 1966 la **Transitron** produce un TTL IC (**Transistor-Transistor Logic Integrated Circuit**) in grado di memorizzare 16 bit di dati.
- Sempre nel 1966 alla IBM viene sviluppata l'idea alla base delle **Dynamic RAM**, usando un solo transistor per memorizzare un bit di dati
- Tra il 1968 e il 1970 alla Fairchild vengono sviluppati circuiti integrati a transistor in grado di memorizzare prima 64 e poi fino a 256 bit di dati.

La nascita della Intel

- Era questo trend che la Intel pensava di sfruttare. Nel 1969 la Intel produce un innovativo chip di SRAM ad alta velocità da 64 bit, il **3101 Schottky bipolar memory**, e nel 1971 la prima **Eeprom** della storia, la **1702**, da 2 Kbit.



- Nel 1970 viene anche prodotto un chip (a causa di alcuni travagli tecnici, solo le versioni consegnate a partire dal 1971 funzioneranno bene) di DRAM, in grado di memorizzare 1024 bit di dati.
- Era l'**Intel 1103**, noto anche come **Magnetic Core Memory Killer**. Per la prima volta una quantità significativa di RAM veniva impacchettata in un unico chip. Nel 1972 diviene il circuito integrato più venduto sul mercato.



La nascita della Intel

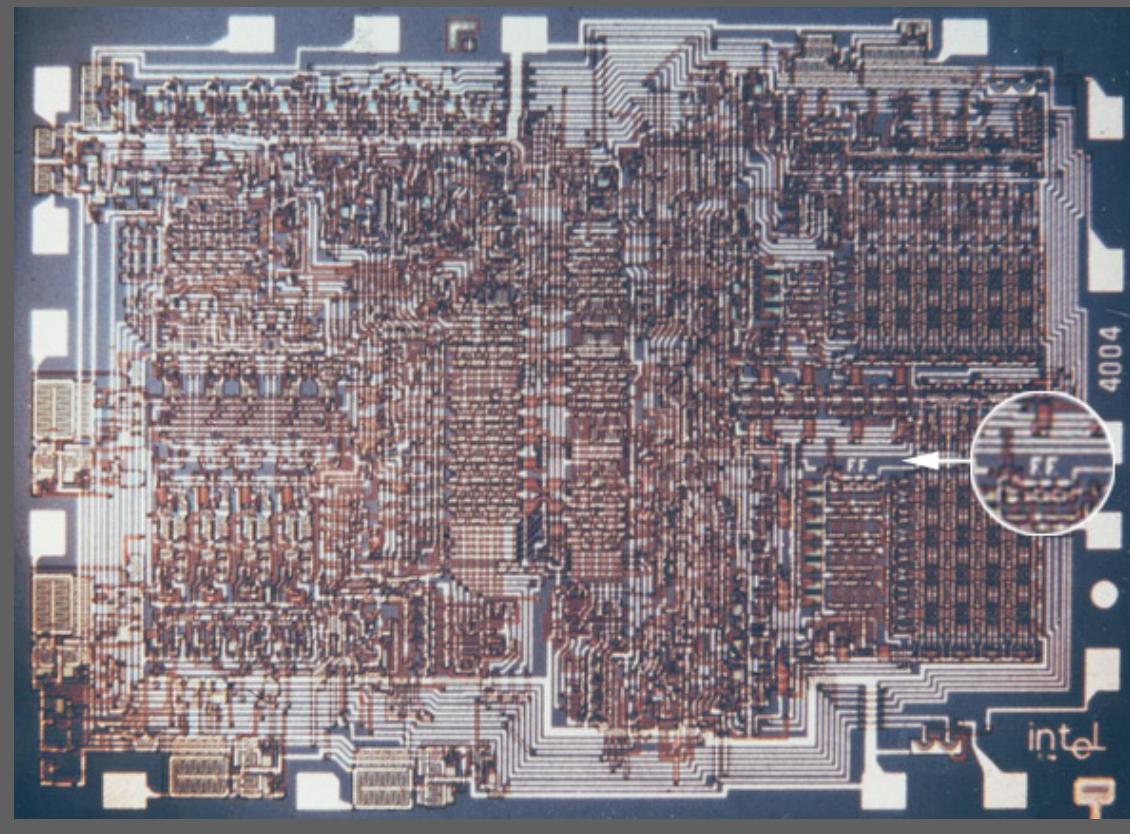
- Intanto, nel 1969 la **Busicom**, una azienda giapponese che produce calcolatrici meccaniche, contatta la appena nata Intel perché vuole costruire una versione elettronica della sua calcolatrice.
- Il progetto ricalcava quello della **Olivetti Programma 101** del 1964, che molti considerano il primo vero personal computer.
- Alla Busicom avevano sviluppato un progetto che richiedeva 12 TTL IC, ma per la Intel produrre 12 diversi IC sarebbe stato notevolmente impegnativo, per cui **Ted Hoff** e **Stan Mazor** mettono a punto un progetto che raccoglie tutti e 12 i circuiti integrati su un unico chip.
- Stimavano infatti che la complessità di questo singolo chip non sarebbe stata molto diversa da quella dei chip di memoria già in fase di progettazione o già in costruzione alla Intel.
- Il chip principale sarebbe poi stato supportato da altri chip per la memoria e per le funzioni di input/output.

La nascita della Intel: Federico Faggin

- Nel 1970 la Intel assume **Federico Faggin** per guidare il gruppo di progettazione e sviluppo di questo nuovo chip, gruppo a cui si era aggiunto **Masatoshi Shima** della Busicom.
- Faggin, nato nel 1941, aveva prima lavorato in Italia per la Olivetti e per la Fairchild, e si era poi trasferito alla sede centrale a Palo Alto nel 1968, dove aveva acquisito una approfondita esperienza nella progettazione di TTL IC basati su nuove tecnologie a semiconduttore.
- Il nuovo chip è pronto per la fine del 1971 ma la Busicom, che era la proprietaria del progetto, non sembra impressionata dal risultato, e non è in grado di garantire l'acquisto di numero sufficiente di pezzi, che avrebbero permesso di tenere bassi i costi di produzione.
- Busicom decide allora ritirarsi dall'affare e di cedere alla Intel i diritti di produzione del nuovo chip in cambio della restituzione del denaro¹⁹¹ che aveva stanziato per il suo sviluppo: 65.000 \$

La nascita della Intel: il 4004

- Faggin decise di chiamare il chip appena sviluppato **4004**: era il primo processore sviluppato completamente su un'unica fettina di silicio.
- Il 4004 era un processore a 4 bit, costruito con tecnologia a 10μ metri, conteneva circa 2.300 transistor, funzionava con un clock di 1 MHz, e eseguiva una media di 60.000 istruzioni al secondo (0,06 MIPS)
- Insieme al 4004 erano stati sviluppati altri tre chip: il 4001(ROM), il 4002 (RAM) e il 4003 (Input/output). Insieme permettevano di assemblare un computer funzionante.
- Faggin firmava con le sue iniziali il suo capolavoro.



La nascita della Intel: l'8008 e l'x86

- Soprattutto grazie alla intelligente scelta del nome, è opinione comune considerare l'8008 l'evoluzione a 8 bit del 4004.
- In realtà invece, furono due progetti indipendenti, che furono portati avanti almeno parzialmente in parallelo.
- Inoltre quella che è probabilmente la più famosa famiglia di CPU col relativo instruction set, insieme noti anche come “**x86**”, il cui capostipite è appunto l' 8008, ha una origine più complessa di quanto si ritenga comunemente.
- Ed è una origine che incomincia al di fuori della Intel, non nel 1972, ma quattro anni prima, nel 1968, lo stesso anno di fondazione della Intel.

Computer Terminal Corporation/Datapoint

- La **Computer Terminal Corporation (CTC)**, poi **Datapoint**, viene fondata nel luglio del 1968 in Texas da **Phil Ray** e **Gus Roche**
- In quel periodo, i computer (mainframe e minicomputer) si usavano collegando ad essi opportuni terminali telescriventi.
- I caratteri digitati venivano stampati sulla carta e inviati al computer. L'output del computer veniva anch'esso stampato sulla carta.
- La telescrivente poteva anche essere dotata di un lettore/perforatore di nastro
- Chiaramente, questi dispositivi erano rumorosi e lenti, a causa del meccanismo di stampa, e l'uso di monitor era molto raro.
- Il modello più diffuso, la **Teletype model 33**, era usata coi computer IBM



Computer Terminal Corporation/Datapoint

- In una intervista del 2004 **Victor Poor**, ingegnere e amico di Roche, dice di essere stato lui, nel 1968, a presentare Ray e Roche ad alcune banche che avrebbero potuto finanziare la nuova compagnia.
- E sarebbe sempre stato Poor a suggerire a Roche l’idea di produrre un terminale più veloce e silenzioso delle normali telescriventi in uso, basato sull’uso di un monitor al posto dell’apparato di stampa (in effetti, un’idea che ha preso piede...)
- Usando alcuni IC prodotti dalla Texas Instr., fu velocemente sviluppato un prototipo di questo nuovo terminale, il **Datapoint 3300** che poteva sostituire la Teletype 33.
- Ovviamente, il “3300” era stato scelto per suggerire che il nuovo terminale era 100 volte più veloce!



Computer Terminal Corporation/Datapoint

- In effetti il 3300 ebbe un immediato successo, ma la CTC non aveva sufficiente capacità produttiva, per cui fu dato in produzione a terze parti, usando chip della Texas Instr. e della Intel.
- Il notevole successo del 3300 sollevò una enorme richiesta di terminali simili, che potessero funzionare con altri modelli di computer, Burroughs, Univac, DEC, e così via.
- Roche allora ricontatta Poor e gli chiede di progettare un nuovo prodotto in grado di emulare il funzionamento di una qualsiasi telescrivente, non solo la model 33.
- Nel frattempo, le azioni della CTC, vendute nell'agosto 1969 a 8 \$, un anno dopo valevano 45 \$ l'una!

Computer Terminal Corporation/Datapoint

- Nell'autunno del 1969 Poor e il suo amico **Harry Pyle** (che al tempo era ancora studente) vengono assunti dalla Datapoint per lo sviluppo del nuovo progetto: un semplice computer in grado di emulare il comportamento di un qualsiasi terminale telescrivente.
- La leggenda (e l'intervista) vuole che Poor e Pyle sviluppino lo schema dell'architettura di base del capostipite di quella che diventerà la famiglia di microprocessori più famosa al mondo, instruction set incluso, sul pavimento del soggiorno della casa di Poor durante i quattro giorni di vacanza per il giorno del ringraziamento, intorno alla fine di novembre del 1969.

Computer Terminal Corporation/Datapoint

- Nelle intenzioni di Poor, l'implementazione della CPU avrebbe potuto essere realizzata usando un centinaio di normali IC TTL **MSI** (**M**edium **S**cale **I**ntegration) già disponibili sul mercato e prodotti dalla Texas Instr. e dalla Intel
- Ad esempio La Texas produceva una serie di IC nota col codice 7400, tra cui la ALU 74181, formata da 75 porte logiche (quindi qualche centinaio di transistor) in grado di eseguire 16 diverse operazioni su due operandi da 4 bit ciascuno.
- Invece Ray e Roche inviano Poor a parlare con la Intel per capire se l'intera implementazione potesse essere realizzata integrando tutto su un unico circuito integrato.
- Così Poor incontra Moore, Noyce, Grove, Mazor e Hoff, i quali però non sembrano molto interessati alla produzione di un unico chip piuttosto esotico, Meglio produrre memorie, pensano, di cui c'è richiesta in grande quantità...
198

Computer Terminal Corporation/Datapoint

- Inoltre, alla Intel stavano sviluppando contemporaneamente il 4004...
- Tuttavia, la Intel era ancora una piccola compagnia, e vendeva la maggior parte delle sue memorie proprio alla Datapoint, per la produzione del 3300, così Phil Ray chiama la Intel per convincerli, altrimenti la Datapoint si sarebbe rivolta ad un'altra azienda per l'acquisto delle memorie.
- Dunque la Intel si decide a produrre questo nuovo chip, intervenendo sul progetto iniziale con alcune modifiche tecniche, sulle specifiche, e mettendo a punto i vari dettagli implementativi.
- Probabilmente, la modifica più importante apportata dalla Intel fu che il progetto iniziale di Poor e Pyle elaborava i byte serialmente, un bit alla volta, mentre nell'implementazione l'informazione scorreva in parallelo su otto bit (in questo aveva aiutato l'aver già lavorato con il 4004)

Computer Terminal Corporation/Datapoint

- Ma alla Intel se la prendono comoda con lo sviluppo del nuovo chip, mentre alla Datapoint hanno fretta di presentare il prototipo del nuovo prodotto ad una convention di banche nell'aprile del 1970.
- Il prototipo viene quindi sviluppato usando normali circuiti integrati (notate, in 6 mesi, a partire dal progetto iniziale del novembre 1969 da G. Asbell, S. Kline, J. Schmidt, J. Register. Nomi ormai dimenticati, ma che forse vale la pena qui di ricordare...).
- Non è del tutto chiaro se alla Datacom fossero consapevoli delle potenzialità del prodotto che stavano sviluppando, non solo un terminale intelligente, ma un vero e proprio **desktop**, un **PC**!
- Ma in ogni caso, per ragioni di marketing, scelsero di presentare sul mercato il nuovo prodotto proprio come terminale intelligente, in grado di collegarsi e dialogare con diversi tipi di computer.

Il Datapoint 2200: un PC in incognito

- E dunque, il **Datapoint 2200** viene commercializzato a partire dal 1970 come terminale programmabile per connettersi a diversi tipi di mainframe.
- Funzionava caricando da nastro magnetico ed eseguendo il programma di emulazione del terminale del mainframe a cui era collegato. Ma se poteva eseguire programmi, era controllabile da tastiera e usava un monitor come output, di fatto era un personal computer!
- E infatti alla Datapoint incominciarono ad arrivare richieste di acquisto da parte di acquirenti che volevano usare il 2200 stand-alone, sviluppando in proprio, in assembler, i programmi di cui avevano bisogno (ad esempio, per buste paga, e per gestire ricette mediche)



Il Datapoint 2200: un PC in incognito

- In effetti, la Datapoint si era rivolta anche alla Texas Instr. per la costruzione del processore in un unico circuito integrato, e alla TI ci avevano anche provato, ma con un risultato troppo scadente, affetto da così tanti malfunzionamenti che non fu mai usato. (non di meno, per molti anni la TI intentò varie cause cercando di sostenere che il brevetto del primo processore su un unico chip apparteneva a loro...)
- Visto il successo del 2200, ne viene presto sviluppata una versione più avanzata, il 2200 II con memoria primaria da 4 KB, estendibile fino a 16 KB, ad un prezzo in valuta attuale da 30.000 a 90.000 \$.
- Intanto, a fine 1971, finalmente alla Intel terminano la produzione del processore commissionato due anni prima, ma alla Datapoint questo ormai serviva poco dato che stanno lavorando su una macchina, il 2200 II appunto, molto più potente.

4004 e 8008 debuttano sul mercato

- Datapoint e Intel (in particolare Ray e Noyce) si mettono d'accordo: l'Intel avrebbe trattenuto i diritti sul nuovo processore (inizialmente denominato 1201), e la Datapoint non avrebbe pagato alla Intel i costi di NRE (Non-Recurring Engineering, i costi di studio e progettazione sostenuti dalla Intel).
- Apparentemente Roche non era molto soddisfatto dell'accordo: Intuiva che l'idea di poter stampare un intero processore su un singolo chip di silicio era rivoluzionaria, e che non avrebbero dovuto cederne la proprietà intellettuale. In seguito avrebbe considerato l'intera vicenda una delle decisioni in affari peggiori della storia.
- Il 4004 viene messo in catalogo nel novembre del 1971, divenendo così il primo microprocessore commerciale della storia. Il 1201, ribattezzato 8008, viene reso disponibile a partire dall'aprile del 1972, venduto al prezzo di 120 \$.

Gli anni '70: l'era dei microprocessori

- A partire dallo sviluppo dei primi microprocessori su chip, è principalmente la tecnologia a giocare un ruolo fondamentale.
- La capacità di inserire componenti elettronici (e quindi porte logiche) sempre più miniaturizzati su un'unica fettina di silicio permette di implementare al meglio alcune delle tecniche ideate già negli anni precedenti, e quindi di costruire processori sempre più:
 - **Veloci**, ossia in grado di lavorare con frequenze di clock sempre più elevate, fino ad arrivare a circa 4 GHz
 - **Sofisticati**, ossia in grado di sfruttare a pieno il parallelismo insito nelle istruzioni del programma in esecuzione, attraverso tecniche come lo **scheduling dinamico della pipeline** e la **rinominazione dei registri**, la **predizione dinamica dei salti**, la **speculazione hardware** e, naturalmente con più memoria a disposizione: **RAM e vari livelli di cache**.

Gli anni '70: l'era dei microprocessori

- La capacità di integrare sempre più transistor su un'unica fetta di silicio cresce velocemente:
- anni '60: **centinaia** di transistor per chip (Medium Scale Integration, **MSI**)
- Primi anni '70: **migliaia** di transistor per chip (Large Scale Integration, **LSI**)
- Anni '80: **centinaia di migliaia** di transistor per chip (Very Large Scale Integration, **VLSI**)
- Anni '90: **milioni** di transistor per chip (Ultra Large Scale Integration, **ULSI**)
- Nel 2005 viene prodotto il primo chip contenente più di un **miliardo** di transistor

Gli anni '70: l'era dei microprocessori

- La capacità di costruire componenti sempre più piccoli viene indicata di solito con il termine “tecnologia a xx micro/nano metri” che indica l’ampiezza di una linea di connessione tra due microcomponenti elettronici.
- I produttori di semiconduttori stimano in realtà che non si possa andare oltre i 10 nm, a causa di alcuni effetti fisici che impedirebbero il corretto funzionamento dei dispositivi.
- Le date future sono ovviamente solo approssimative (Fonte: wikipedia)

10 μm — 1971
3 μm — 1975
1.5 μm — 1982
1 μm — 1985
800 nm (0.80 μm) — 1989
600 nm (0.60 μm) — 1994
350 nm (0.35 μm) — 1995
250 nm (0.25 μm) — 1997
180 nm (0.18 μm) — 1999
130 nm (0.13 μm) — 2002
90 nm — 2004
65 nm — 2006
45 nm — 2008
32 nm — 2010
22 nm — 2012
14 nm — 2015
10 nm — est. 2017₂₀₆?
7 nm — est. 2020 ???

Gli anni '70: l'era dei microprocessori

- La frequenza del clock dei microprocessori passa dai 0,8 MHz del 8008 nel 1972 ai quasi 4 GHz del Pentium 4 nel 2000 (ma a metà degli anni '60 l'IBM 7030 aveva già un clock di 1 MHz, e il CDC 6600 funzionava con un clock di 10 MHz. Il 4004 usava un clock da 1 MHz).
- La RAM indirizzabile dal processore passa da qualche Kbyte a un valore virtualmente quasi illimitato, per i processori a 64 bit, e anche i computer di fascia bassa hanno almeno qualche Gbyte di RAM.
- I processori vengono progressivamente dotati di più livelli di cache, fino ai 3 attuali, con una cache di terzo livello di qualche Mbyte.
- La dimensione *naturale* dei dati manipolati dal processore passa, progressivamente, da 8 a 64 bit.

Gli anni '70: l'era dei microprocessori

- Naturalmente, non mancano anche le innovazioni architetturali, su cui torneremo più avanti. Le più importanti sono:
- La **rivoluzione RISC** all'inizio anni '80, con cui viene drasticamente rivista la natura delle istruzioni macchina, permettendo così di costruire macchine più semplici, più veloci, e più adatte allo sfruttamento del pipelining.
- Lo sfruttamento intensivo dell'**Instruction Level Parallelism** negli anni 90.
- L'introduzione del **multithreading** e poi dei **processori multi-core** tra la fine degli anni '90 e i primi anni del 2000
- L'uso dei processori grafici per elaborazione non grafica, indicati con le sigle **GPGPU**, o **GPGP**, o **GP²U**, a partire dall'inizio degli anni 2000.

Xerox PARC Alto PC

- Nel 1973 nei laboratori di ricerca della **Xerox**, a Palo Alto , viene sviluppato **Alto**, il primo desktop/personal computer dotato di mouse, interfaccia grafica e collegato ad una rete locale ethernet.
- Ebbe una commercializzazione limitata, ma ne furono prodotti alcune migliaia di esemplari, usati alla Xerox e in varie università.
- Chiaramente ispirato alle idee di Engelbart, era sostanzialmente un minicomputer, ma dato che veniva usato da una singola persona alla volta seduta al tavolo su cui era poggiata la tastiera, venne naturale considerarlo un **Personal Computer**
- Influenzò tra gli altri i progetti del Macintosh e delle workstation Sun (ci torneremo parlando dei linguaggi e dei sistemi operativi) 209

1976: nasce la Apple

- **Steve Jobs** e **Steve Wozniak**, entrambi californiani si erano conosciuti e frequentati per un po' di tempo nel 1971, quando Jobs aveva 16 anni e Wozniak 21.
- Nel 1975 entrambi lasciano i rispettivi college, senza laurearsi. Wozniak era un genio autodidatta dell'elettronica, aveva già costruito un dispositivo elettronico per telefonate a lunga distanza a costo zero (e ne aveva venduti alcuni centinaia di pezzi), e un terminale video per collegarsi al minicomputer del college, di cui erano stati venduti alcuni esemplari.
- Apparentemente, fu Jobs a spingere Wozniak ad interessarsi di più ai computer, sicché nel marzo del 1976 Wozniak incomincia a partecipare agli incontri di un club locale di appassionati di computer: la sua intenzione era di trasformare il terminale video che aveva progettato qualche anno prima in un vero computer.

1976: nasce la Apple

- Ma i microprocessori disponibili in quel momento, l'Intel 8080 e il Motorola 6800 costavano troppo per le sue tasche (170 dollari del tempo, ossia circa 7/800 \$ attuali), per cui il suo progetto, basato sul 6800, rimane inizialmente su carta.
- Presto però compare sul mercato il 6502, un microprocessore supereconomico con caratteristiche molto simili al 6800 (era infatti stato progettato dai alcuni ex-dipendenti della Motorola), e Wozniak costruisce velocemente il suo piccolo computer e lo mette in mostra al club locale di hobbysti di computer.
- Qui reincontra Jobs, che frequentava lo stesso club e intuiva un potenziale commerciale nelle “macchinette” che i vari appassionati appartenenti al club costruivano per hobby.



1976: nasce la Apple

- Jobs riesce in pochissimo tempo a trovare un acquirente, un negozio locale di computer, il Byte shop, disposto ad acquistare 50 esemplari del computer appena sviluppato da Wozniak a 500 \$ l'uno (circa 2000 \$ attuali), che verrà poi venduto col nome di **Apple I**.
- E convince un distributore di componenti elettronici a vendergli a credito i pezzi necessari a costruire i 50 esemplari: lo pagherà a 30 giorni, quando riceverà i soldi dal Byte Shop.
- Jobs e Wozniak, con alcuni aiutanti, lavorano giorno e notte per assemblare i 50 esemplari in un mese, a cui ne seguiranno altri (per un totale di circa 200) sempre costruiti con pezzi presi a credito.



1976: nasce la Apple

- Intanto, Jobs, Wozniak e **Mike Markkula** fondano, il primo di aprile del 1976 la **Apple Computers Inc.**
- Markkula era il classico imprenditore americano, sempre in cerca di nuove idee da finanziare che si fa convincere da Jobs a finanziare la Apple con 250 mila \$, garantendo con il suo nome e le sue capacità di imprenditore i crediti bancari e gli investimenti necessari per lo sviluppo della attività appena nata.
- Il nome “Apple” viene scelto perché, sull’elenco telefonico, Apple sarebbe comparso prima di Atari (e apparentemente perché Jobs ricordava con piacere un suo precedente lavoro estivo in una fattoria che produceva mele in Oregon...)



1976: nasce la Apple

- Intanto, Wozniak si dedica già alla progettazione dell'**Apple II**, che viene presentato ad una esposizione di computer nell'aprile 1977.
- Jobs insiste sul fatto che il computer dovrà funzionare appena estratto dalla scatola, senza dover configurare nulla né collegare altri dispositivi che non siano la presa di corrente.
- Segue nel 1980 l'**Apple III**, un computer simile al II, ma con un grave difetto, la mancanza di ventilatore, sempre su idea di Jobs, che però costringerà la Apple a ritirare dal mercato circa 14.000 esemplari.
- La Apple subirà un notevole danno di immagine, proprio mentre sta entrando sul mercato dei personal computer la IBM



1981: il PC IBM

- Il 12 agosto del 1981 la IBM fa la sua comparsa sul mercato dei personal computer con l'**IBM 5150**, un computer da tavolo (appunto, un desktop computer) con processore Intel 8088 a 4,77 MHz, da 16 a 256 Kbyte di RAM e Floppy Disk per la memoria di massa.
- Il termine Personal Computer era già comparso almeno a partire dall'Alto della Xerox, ma data la rapida diffusione del 5150, presto l'acronimo PC diventa sinonimo di “*computer compatibile con i personal computer IBM*”
- In poco tempo, produttori di periferiche, schede di espansione e software compatibili con i PC IBM cominciano a proliferare, formando così una enorme disponibilità di prodotti standardizzati sulla piattaforma IBM.
- “**IBM compatibile**” diventa una qualità irrinunciabile per garantire il successo sul mercato, a cui solo i prodotti Apple riescono a resistere

PC vs MAC

- Secondo Bill Gates, che nell'agosto del 1981 si trovava in visita proprio alla Apple, i dirigenti dell'azienda non sembravano rendersi conto di cosa potesse significare per loro l'ingresso della IBM sul mercato del personal computer
- A fine agosto la Apple acquista sul Wall Street Journal una pagina di pubblicità che recita: “*Welcome IBM, seriously*”
- Ma il tono di sufficienza si rivela fuori luogo: nel giro di tre anni la IBM detiene il 56% del mercato dei personal computer negli USA, contro il 16% della Apple.



PC vs MAC

- Nel frattempo, a fine 1979, mentre era ancora in produzione l'Apple III, dietro pagamento di un milione di dollari in azioni, ad alcuni dirigenti Apple viene concesso di visitare per tre giorni Xerox Parc.
- Da qui nascerà l'idea di produrre un nuovo computer dotato di interfaccia grafica, con finestre e icone manipolabili da un mouse. Inizialmente il **LISA**, con scarso successo perché troppo costoso, ma subito dopo seguito dal **Macintosh 128k**, lanciato sul mercato con una imponente campagna pubblicitaria a partire dal 22 gennaio 1984.
- Il nuovo computer della Apple ha subito un enorme successo, e segna l'inizio della definitiva divisione dell'universo dei personal computer in due grandi mondi, incompatibili fra loro per molti anni: Mac e PC



Gli anni ‘80: the RISC Revolution

- Intanto, **David Patterson** (all'università di Berkley), e **John Hennessy** (all'università di Stanford) incominciano a condurre i loro esperimenti su un nuovo modello architettonale e su un nuovo concetto di instruction set architecture.
- Negli anni '80, i risultati di questi esperimenti daranno luogo alla **RISC** revolution, un nuovo tipo di processori caratterizzati da istruzioni macchina semplici, struttura regolare e dimensione fissa (appunto: **Restricted Instruction Set Computer**)
- Sono le nuove macchine RISC, che nel giro di 10 anni sostituiranno i processori progettati fino a quel momento, e che verranno indicati retrospettivamente con l'acronimo **CISC** (**Complex Instruction Set Computer**), presto sinonimo di “macchina obsoleta”.

Complex Instruction Set Computers: CISC

- Negli anni ‘60 e ‘70, la definizione dell’instruction set (ISA) della maggior parte dei processori aveva seguito linee guida ben precise, dettate dalla tecnologia del tempo e dal grado di sviluppo dei linguaggi di programmazione e dei relativi compilatori.
- Quasi tutti i processori impiegavano instruction set con le seguenti caratteristiche:
 1. Un gran numero di istruzioni macchina
 2. Istruzioni macchina complesse, ossia capaci esprimere molto lavoro
 3. Molteplici modalità di indirizzamento della memoria dati
 4. Possibilità per tutte le istruzioni di indirizzare la memoria dati usando una delle tante modalità di indirizzamento a disposizione.

Complex Instruction Set Computers: CISC

- Un tipico esempio di queste caratteristiche era l’istruzione MVC dell’IBM 360, che permetteva di spostare fino a 256 byte da un punto all’altro della memoria primaria. L’IBM 370 aveva anche l’istruzione MVCL che permetteva di usare stringhe più lunghe di 256 byte.
- La MVC poteva essere usata con diverse modalità di indirizzamento, anche piuttosto complesse. Ad esempio:
 - MVC 15 (100, 5), 50 (8)
- Ossia: sposta 100 byte consecutivi che trovi a partire dall’indirizzo in memoria dato da $50 + [R8]$ all’indirizzo $15 + [R5]$
- L’istruzione quindi specifica in modo molto sintetico l’equivalente di due addizioni, 100 LOAD e 100 STORE!

Complex Instruction Set Computers: CISC

- Istruzioni macchina complesse permettevano di esprimere molto lavoro in modo estremamente conciso, il che in effetti forniva un certo numero di vantaggi:
 1. Permetteva di generare eseguibili corti, che occupavano cioè poco spazio in memoria principale, che negli anni ‘60 e ‘70 aveva normalmente dimensioni molto limitate.
 2. I tempi di accesso alla memoria principale erano elevati, dato che le memorie cache non erano ancora usate. Ogni volta che si prelevava una istruzione dalla memoria, aveva dunque senso che questa esprimesse una gran quantità di lavoro.
 3. La tecnologia dei compilatori era ancora in fase di sviluppo, e l’uso di istruzioni macchina complesse semplificava la loro progettazione.

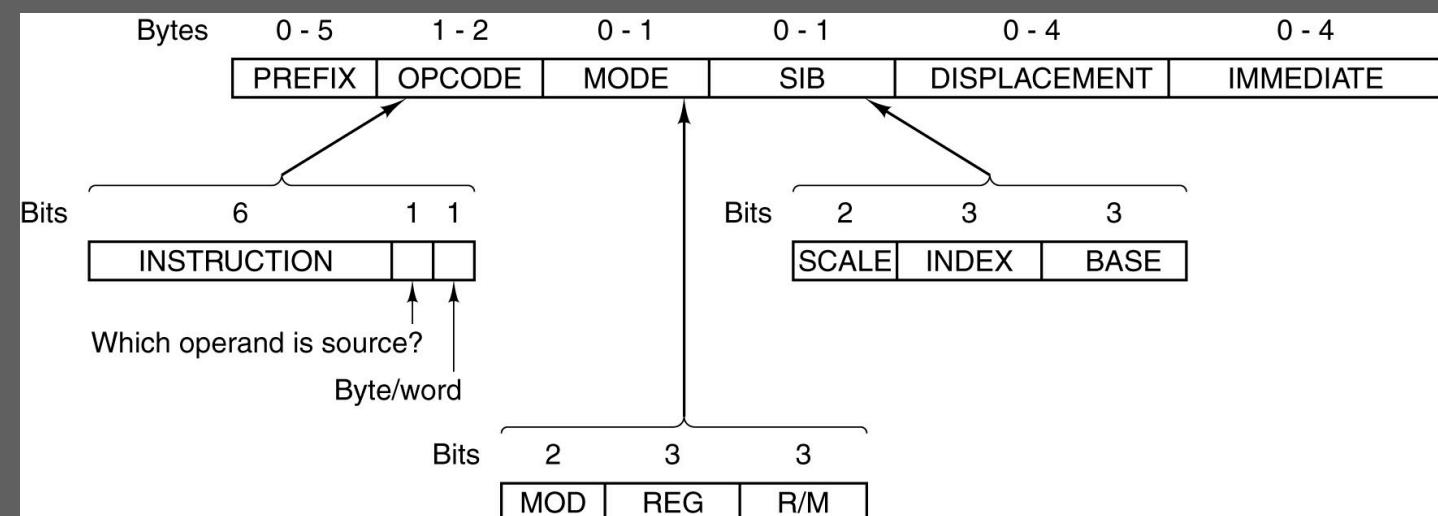
Complex Instruction Set Computers: CISC

- Istruzioni complesse riducevano quindi il gap semantico tra un programma scritto in un linguaggio ad alto livello e la sua traduzione in linguaggio macchina.
- Infatti, parte della complessità della traduzione veniva trasferita all'interno del processore: se ne faceva carico il microprogramma necessario all'esecuzione delle istruzioni macchina.
- L'uso della microprogrammazione contribuì a sua volta a generare instruction set sempre più complessi e articolati.
- Poiché il microprogramma era memorizzato su ROM, veniva comodo arricchire l'instruction set di una CPU con nuove istruzioni macchina semplicemente aggiungendo il relativo microprogramma alla ROM.
- Si poteva addirittura cambiare l'intero instruction set di una CPU cambiando solo la ROM che conteneva i microprogrammi!

Complex Instruction Set Computers: CISC

- Ma tutta questa versatilità aveva delle conseguenze:
- Istruzioni macchina complesse richiedevano molti byte di descrizione, che però non servivano per le istruzioni più semplici. Per non sprecare inutilmente spazio, le istruzioni dovevano avere lunghezza variabile.
- Ad esempio, nell'ISA x86 le istruzioni sono lunghe da 1 a 17 byte.
- La lunghezza variabile delle istruzioni faceva sì che, al loro interno, i bit usati per specificare gli operandi si trovassero in posizioni diverse, a seconda del tipo di istruzione.

- Il formato delle istruzioni x86 nel Pentium 4



Complex Instruction Set Computers: CISC

- Istruzioni fatte così hanno elevati tempi di esecuzione. Infatti:
- La memoria di istruzioni viene letta a blocchi di byte di dimensione fissa, e una volta prelevati, non si sa ancora se contengono la parte centrale di un'istruzione, o la parte finale di un'istruzione e la parte iniziale dell'istruzione successiva, o due istruzioni intere, e così via.
- Una volta individuata una istruzione, prima di poterne prelevare gli operandi occorre aver decodificato il suo codice operativo, per sapere in quali bit sono specificati gli operandi stessi.
- Istruzioni complesse richiedono datapath e control unit più complesse, (ossia fatti di più porte logiche) e quindi tempi di commutazione più lunghi (ossia un ciclo di clock più lungo) o pipeline più profonde.
- Poiché tutte le istruzioni possono indirizzare la RAM, questa ²²⁴ diviene facilmente un collo di bottiglia!

Resctricted Instr. Set Computers: RISC

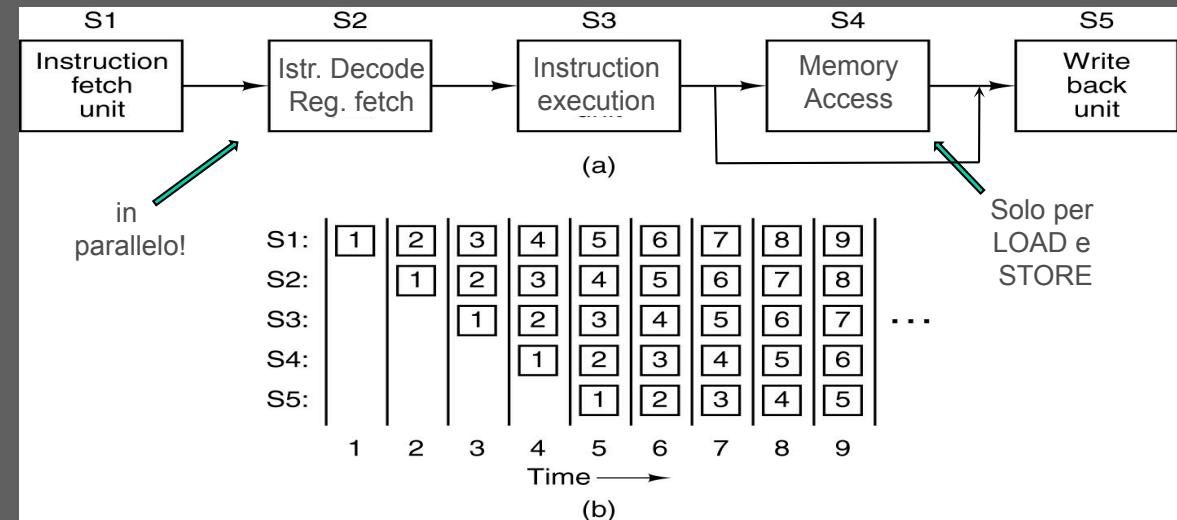
- Mentre la commercializzazione di processori con ISA complessi va avanti, tra la fine degli anni ‘70 e l’inizio degli anni ’80 alcuni ricercatori incominciano a sperimentare un nuovo paradigma architetturale.
- Questi esperimenti affondano le radici nei lavori degli anni ‘60 di Seymour Cray con i CDC 6600/6700 e in una macchina sperimentale dell’IBM, l’801 sviluppata negli anni ‘70 ma mai commercializzata.
- A **Berkeley**, D. Patterson e C. Sequin lavorano su un nuovo processore che evolverà poi nelle famiglie di processori commerciali **SPARC** (e coniano gli acronimi RISC e CISC)
- A **Stanford**, J. Hennessy sviluppa un nuovo processore chiamato **MIPS**, che evolverà nella famiglia di processori MIPS usati soprattutto per applicazioni embedded.

Resctricted Instr. Set Computers: RISC

- Una architettura RISC impiega un numero ristretto di **istruzioni macchina semplici**, che specificano cioè poco lavoro, come la somma o la sottrazione di due registri, e poco altro.
- Per questa ragione il datapath è corto, un automa a stati finiti è sufficiente per descrivere la control unit (quindi niente microprogrammi), e si possono usare cicli di clock più brevi.
- **Le istruzioni hanno tutte la stessa lunghezza**, ad esempio 32 bit. Ogni accesso alla memoria istruzioni preleva esattamente una o più istruzioni complete.
- **Le istruzioni hanno struttura regolare**: in particolare gli operandi sono specificati sempre dagli stessi bit, per cui gli operandi possono essere prelevati prima ancora di sapere se e come verranno usati.
- **Solo LOAD e STORE possono indirizzare la RAM**, le altre istruzioni usano registri. Quindi, niente colli di bottiglia verso la RAM

Restricted Instr. Set Computers: RISC

- Queste caratteristiche rendevano le macchine RISC particolarmente adatte all'esecuzione in pipeline delle istruzioni: la loro semplicità faceva sì che ogni fase della pipeline non interferisse con le altre.
- MIPS: **Microprocessor without Interlocked Pipelines Stages**
- SPARC: **Scalable Processor ARChitecture**
- Il formato delle istruzioni della architettura SPARC a 32 bit



Format	2	5	6	5	1	8	5	
1a		DEST	OPCODE	SRC1	0	FP-OP	SRC2	3 Register
1b		DEST	OPCODE	SRC1	1	IMMEDIATE CONSTANT		Immediate
2	2	5	3			22		
		DEST	OP			IMMEDIATE CONSTANT		SETHI
3	2	1	4	3		22		
		A	COND	OP		PC-RELATIVE DISPLACEMENT		BRANCH
4	2				30			
					PC-RELATIVE DISPLACEMENT			CALL

RISC vs CISC

- Rispetto alle macchine CISC quindi, una macchina RISC usa codice macchina più semplice, e i programmi RISC sono più lunghi e usano più registri. Ad esempio, l'istruzione del VAX:
 - ADDL3 42 (R1), 56 (R2), 0 (R3)
- è lunga 10 byte e usa 3 registri. In una macchina RISC corrisponderebbe a una sequenza di istruzioni simile a questa:
 - LD R4, 56 (R2)
 - LD R5, 0 (R3)
 - ADD R6, R4, R5
 - SD R6, 42 (R1)
- Che occupa in tutto 16 byte e usa 6 registri.

RISC vs CISC

- Rispetto ai processori CISC dunque, per funzionare adeguatamente il nuovo modello architetturale aveva bisogno di più memoria RAM e di un numero adeguato di registri architetturali (ossia i registri nominabili al livello ISA)
- Del resto, tra la fine degli anni '70 e l'inizio degli anni '80, la tecnologia a semiconduttori per la costruzione delle memorie RAM aveva ormai completamente sostituito le vecchie memorie a nucleo magnetico, e la dimensione media della memoria principale andava crescendo costantemente.
- Mentre le macchine CISC commercializzate in quel periodo avevano da 8 a 16 registri, la MIPS nasceva già con 32 registri a livello ISA.

RISC vs CISC

- L'argomento dei RISC supporter era semplice: è vero che una macchina RISC ha bisogno di 4 o 5 istruzioni per fare ciò che una macchina CISC riesce a fare con una sola istruzione, ma se le istruzioni RISC sono 10 volte più veloci di quelle CISC, RISC vince.
- In breve ne nasce una sorta di guerra di religione, che vede i sostenitori del nuovo paradigma RISC attaccare l'ordine costituito, rappresentato dai grandi produttori di macchine CISC: la Digital Equipment Corporation (DEC), la Intel e l'IBM.
- Già nel settembre del 1980 D. Clark e W. Strecker, progettisti del VAX, pubblicano un articolo che inizia così:

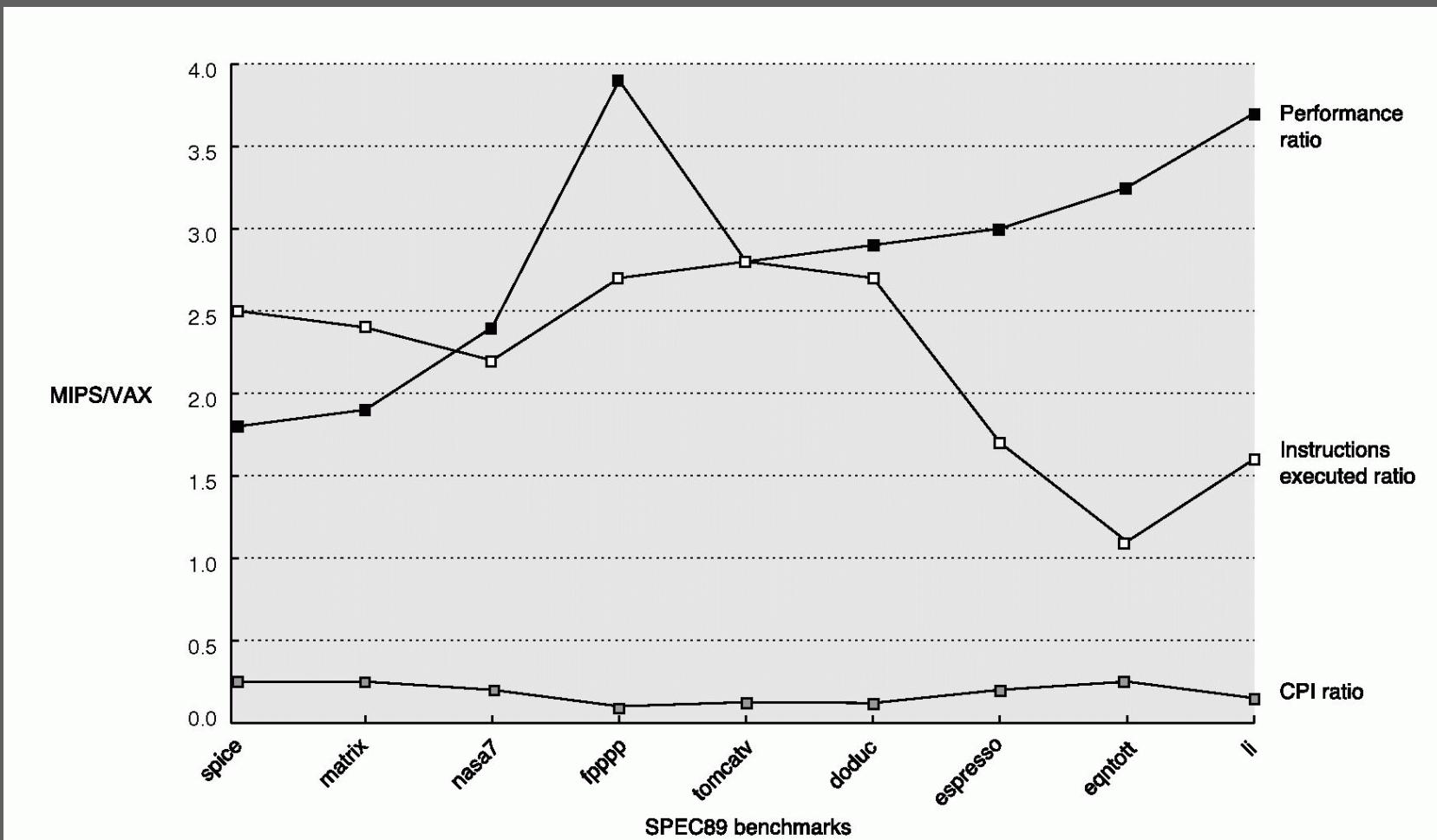
The superiority of a RISC over a CISC will be very difficult to prove. We suggest that several of Patterson's points are misleading, and we present some evidence on the other side of the argument. We rely heavily, as did they, on the VAX-11 architecture for examples. 230

RISC vs CISC

- Non di meno, a partire dai primi anni ‘80, l’idea RISC prende piede, e alcune compagnie incominciano a produrre processori basati su questo nuovo modello architetturale.
- Ma per sapere chi aveva ragione era necessario mettere a confronto una macchina CISC e una RISC, cosa che si poteva fare solo nel momento in cui fosse stato disponibile una macchina RISC sufficientemente potente. Il confronto venne fatto nel 1989, tra il VAX 8700 (CPU: REX520) e la MIPS M2000 (CPU: MIPS R3000).
- La serie VAX della DEC era la vera quintessenza di macchina CISC. Successore dei PDP-11, prevedeva l’ortogonalità totale dell’ISA: ogni istruzione poteva usare ogni modalità di indirizzamento dei dati.
- La MIPS M2000 aveva caratteristiche hardware molto simili al VAX 8700, per numero di transistor del processore, frequenza del clock e dimensione della cache.

RISC vs CISC

- La MIPS eseguiva in media il doppio delle istruzioni del VAX, ma una istruzione MIPS veniva eseguita in 1/6 del tempo necessario all'esecuzione di una istruzione del VAX. Ossia, **la MIPS era mediamente tre volte più veloce del VAX!**



RISC vs CISC

- Il risultato era sufficientemente chiaro, e dall'inizio degli anni '90 in poi tutti i produttori di processori si spostano verso la progettazione di architetture RISC, inclusa la DEC con la architettura **Alpha**.
- Va detto che col tempo una delle caratteristiche RISC, la semplicità delle istruzioni, è un po' venuta meno, ma sono rimaste le altre: istruzioni di lunghezza fissa e struttura regolare, accesso limitato alla RAM, sfruttamento facilitato del pipelining.
- Apparentemente, solo l'Intel ha resistito alla rivoluzione, dato che l'ISA dei processori Intel è ancora chiaramente CISC, ma...
- A partire dal 486 (1989) i processori Intel contengono un cuore RISC: le istruzioni *x86*, tipicamente CISC, superata la fase di fetch, vengono tradotte in una o più microistruzioni (dette *uops*) di struttura regolare e lunghezza fissa di 72 bit. Sono queste uops ad essere avviate verso la fase di esecuzione (fase EXE della pipeline a 5 stadi) vera e propria.

Gli anni ‘90: Instruction Level Parallelism

- È negli anni 90 che raggiungono la loro massima espressione le tecniche di **Instruction Level Parallelism (ILP)**: lo sfruttamento del parallelismo insito nelle istruzioni del programma in esecuzione:
- **Scheduling dinamico della pipeline**: le istruzioni indipendenti fra loro vengono eseguite in un qualsiasi ordine, in modo da non provocare stall della pipeline.
- **Branch prediction dinamico**: l'esito dei salti condizionati viene predetto sulla base delle esecuzioni precedenti, in modo da sapere prima possibile con quale istruzione prosegue la computazione.
- **Speculazione hardware**: l'esecuzione di istruzioni prima ancora di sapere se queste devono effettivamente essere eseguite.
- **Multiple issue**: l'avvio all'esecuzione di più istruzioni in parallelo.
- **Ottimizzazione del compilatore**: per produrre codice che permetta alla CPU di sfruttare al meglio il parallelismo insito in un programma.²³⁴

Anni 2000: CPU multithreaded e multi-core

- alla fine degli anni ‘90 diviene chiaro che si sono più o meno raggiunti i limiti alla capacità di costruire processori sempre più in grado di sfruttare il parallelismo insito nelle istruzioni di un programma.
- Per aumentare ulteriormente le prestazioni l'unica strada percorribile è quella di esplicitare il parallelismo delle applicazioni, e di eseguire il codice su processori in grado supportare questo parallelismo.
- I **processori multithreaded** sono in grado di eseguire in parallelo istruzioni appartenenti a più peer thread.
- I **processori multi-core** (da due core in su) permettono di eseguire in parallelo il codice di più processi.
- Le due tecniche vengono combinate, ed ormai normale l'uso di processori multi-core in cui ciascun core implementa il multithreading

Anni 2000: General Purpose GPU

- Negli anni 2000 inizia anche l'uso dei processori grafici (GPU: Graphic Processing Unit) per elaborare informazione non grafica.
- Le GPU possono elaborare in parallelo grandi quantità di dati in virgola mobile, fornendo quindi prestazioni molto elevate in quei campi che ne facciano uso, come per le previsioni del tempo, la simulazione di fenomeni fisici e altre applicazioni scientifiche.
- I supercomputer più potenti al mondo sono ormai costituiti da milioni di core, e spesso combinano assieme CPU multi-core e GPU per raggiungere un potere computazionale di migliaia di **TeraFLOPS**, ossia milioni di miliardi di **OPerazioni FLoating points al Secondo**.

prestazioni delle CPU dal 1978 al 2010

