a1.6 SEKVENCIJALNI SISTEMI

Kod sekvencijalnih mreža, trenutna vrednost logičke funkcije nije uslovljena samo trenutnom vrednošću ulaznih logičkih promenljivih već i vrednostima koje su one imale u prethodnim trenucima vremena odnosno, vremenskim sekvencama. Sekvencijalne mreže memorišu svoja prethodna stanja i koriste ih u izvodjenju vrednosti trenutne izlazne logičke funkcije. Sekvencijalne mreže su dinamički sistemi.

Osnovni logički elemenat koji može da memoriše svoje stanje je bistabilni multivibrator koji se izvodi u dva osnovna oblika RS leč i RS flip-flop.

a1.6.1 Bistabilni logički elementi

RS flip-flop je regenerativna logička struktura koja poseduje dva stabilna stanja – logička nula i logička jedinica. Inicijalno stanje RS filpflopa je logička nula – resetovano stanje. Dovodjenjem pobude na S ulazni kanal RS flip-flop se prevodi u SET stanje a dovodjenjem pobude na R ulazni kanal RS flip-flop se prevodi u RESET stanje. SET stanju odgovara logička vrednost jedinice a RESET stanju odgovara logička vrednost nule. Logička tabela RS flip-flopa prema tome glasi:

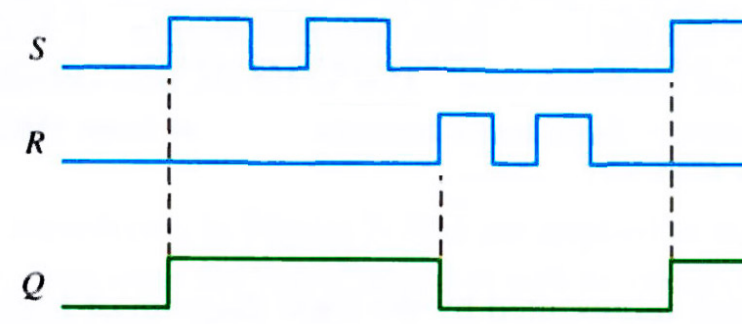
R(t)	S(t)	Q(t)	Q(t+1)	
0	0	0	0	Inicijalno stanje
0	1	0	1	**Set funkcija**
0	0	1	1	
0	1	1	1	
1	0	1	0	**Reset funkcija**
0	0	0	0	
1	0	0	0	
1	1	0	-	Nedozvoljena stanja
1	1	1	-	

Ako RS flip-flop posmatramo kao memorijski elemenat, onda je ova sekvenca UPIS u memoriju logičke jedinice

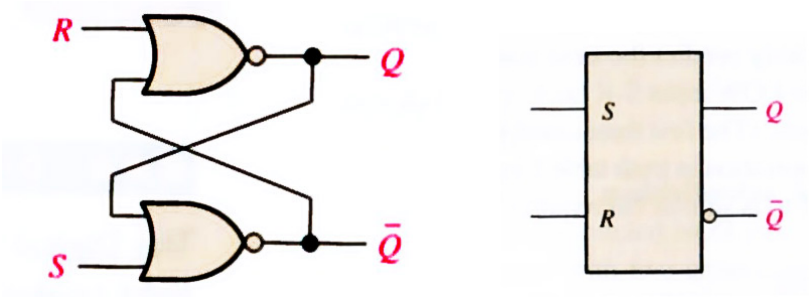
Ako RS flip-flop posmatramo kao memorijski elemenat, onda je ova sekvenca BRISANJA, odnosno UPIS u memoriju logičke nule

Logika rada RS flip-flopa je identična radu bistabilnog električnog preklopnika, odnosno prekidača na zidu, kojim uključujemo i isključujemo svetlo.

Grafički prikaz funkcije RS flip-flopa navodi se na slici ispod.



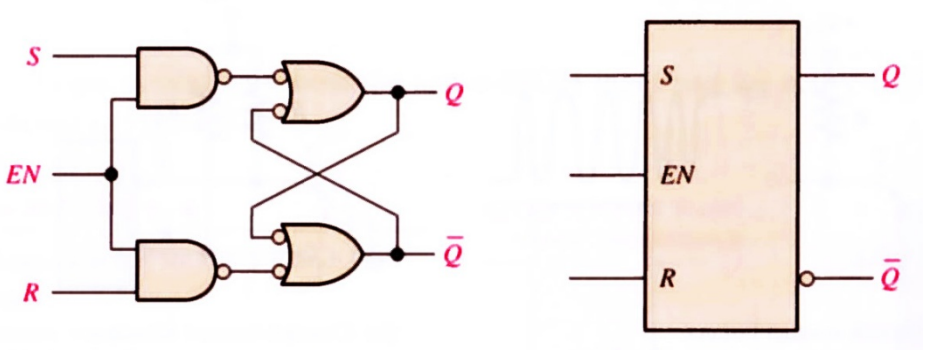
Logički dijagram i grafički simbol prikazan je na slici ispod.



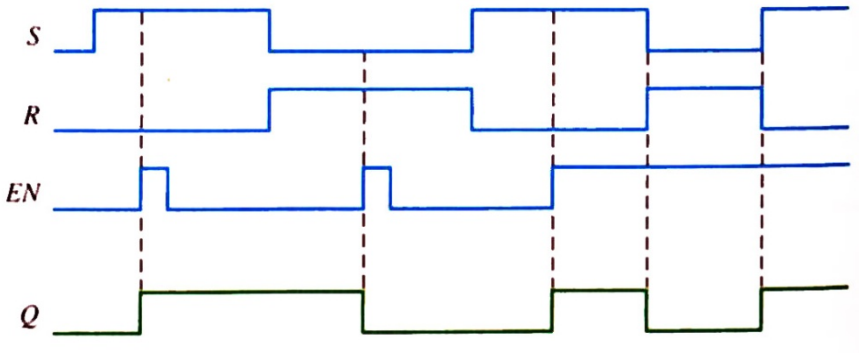
Flip-flop se sreće u različitim tehničkim izvodjenjima. Pored osnovnog RS izvodjenja, ugradnjom specifičnih električnih kola na njegovim ulazima dobijaju se: taktovani flip-flop koji funkcioniše u sinhronizmu sa nekim izvorom taktnih impulsa, T flip-flop koji ima jedan pobudni ulaz a svoje stanje na izlazu menja u zavisnosti od prethodnog izlaznog stanja (funkcija ekvivalentna mehanizmu za aktuiranje hemijske olovke), JK filpflop u kome je objedinjena funkcija RS i T flip-flopa, D flip-flop, ...

Taktovani (sinhroni) filpflop

Taktovani (sinhroni) filpflop, logička šema i grafički simbol. Set i reset ulazni kanali su preko logičkog I kola spregnuti sa spoljašnjim izvorom koji preko EN kanala daje dozvolu za prelazak stanja RS flip-flopa. Ukoliko je spoljašnji izvor neki generator taktnih impulsa, RS filp-flop će raditi u sinhronizmu sa generatorom takta.

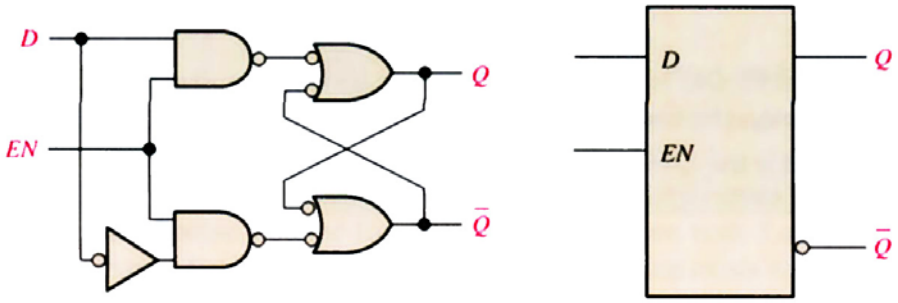


Princip funkcionisanja T flip-flopa prikazan je na sledećem dijagramu.

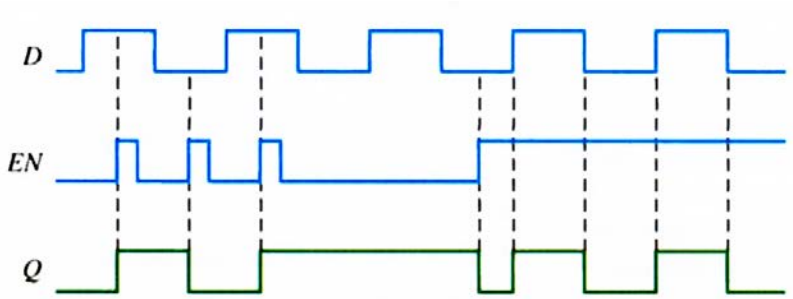


D flip-flop

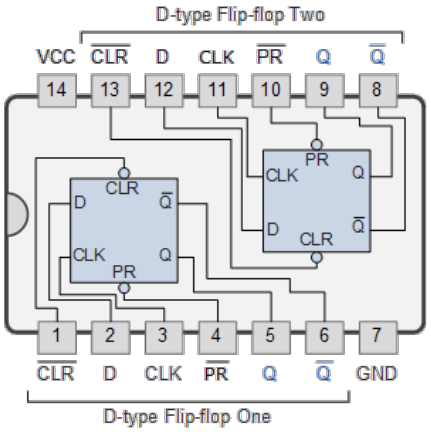
D flip-flop rešava problem pojave zabranjenog stanja, odnosno istovremenog aktiviranja R i S ulaza. Invertovanjem S kanala i dovodjenjem na R ulaz filp-flopa invertovanog signala, ostvaruje se funkcija ovog memorijskog elementa **samo preko jednog ulaza** koji ima istovremeno funkciju set i reset, u zavisnosti od stanja u kome se sklop nalazi (mehanizam hemijske olovke, toglovanje). Prelaz u komplementarno stanje ostvaruje se uvek na rastuću ivicu EN kanala. Ovaj flip-flop nosi naziv od engleske reči *delay* (kašnjenje) zato što izlazni signal kasni u odnosu na ulazni. Jedan komandni ulaz omogućava efikasnu primenu kod izgradnje memorijskih elemenata, pa zato odatle dolazi i njegov drugi naziv – *data* flipflop.



Princip funkcionisanja D flip-flop je prikazan na sledećem dijagramu.

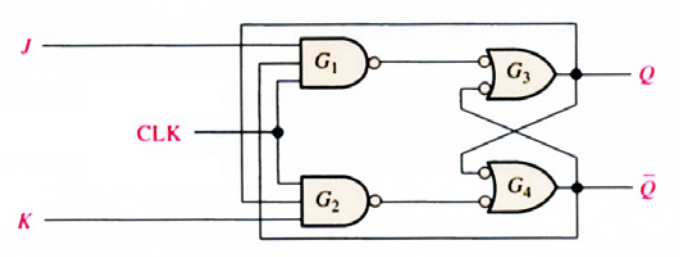


74LS74 Dual D-type Flip Flop

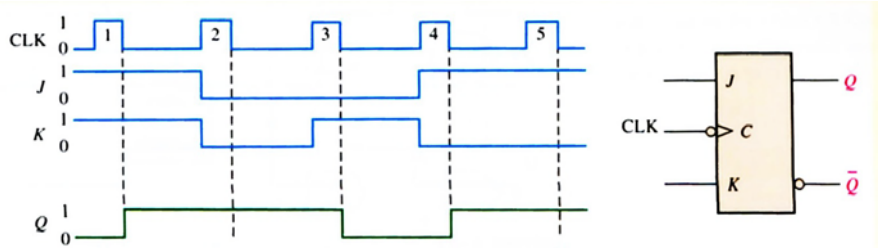


JK flip-flop

Potpuno razrešenje problema pojave nedozvoljenih stanja na ulazu RS flip-flopa ostvareno je kroz dodatne povratne sprege. Ovaj oblik RS flip-flopa naziva se **JK flip-flop**. Logička šema JK flip flopa prikazana je na sledećoj slici.

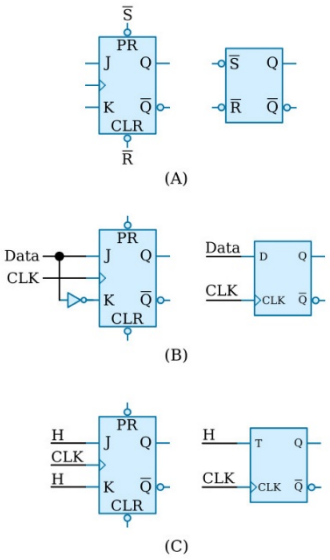


Taktni dijagram prikazan je na narednom dijagramu. Grafički simbol je takodje prikazan na sledećoj slici. Može se uočiti da se umesto R i S oznaka na ulazu, koriste simboli J i K u čast Jack Kilby-ja koji se smatra pronalazačem integrisanih kola, koja su tehnološka osnova modernih informacionih tehnologija i mehatronskih sistema.



Praktična primena memorijskih elemenata omogućava realizaciju različitih sekvencijalnih logičkih struktura koje se sreću kao funkcionalni moduli savremenih digitalnih računara. U okviru ovog nastavnog modula biće obradjene sekvencijalne strukture koje nazivamo registrima i brojačima.

JK je fleksibilan i može se konfigurisati tako da zameni druge vrste flipflopova.



Dodatno pojašnjenje:

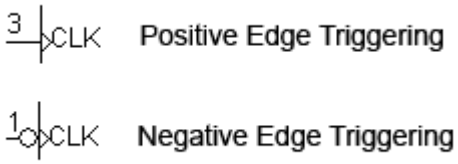
Flip-Flops and Latches

Flip-flops are a useful type of digital device that can store binary states, or be used as a sort of digital toggle switch. Flip-flops can be built up from distinct logic gates, but they can easily be bought in packaged chips. The flip-flop is a digital device, so its output (labeled with a *Q* in schematic diagrams) will take on the value of either 1 (high) or 0 (low). Many flip-flops will also provide the **complementary** output (an inverted output).

Edge Triggering

Because the state of a flip-flop often depends on the previous state of a circuit (for example, the output of one flip flop may be the input to another), and because each flip-flop and logic gate needs a certain amount of time to switch its output, we usually **clock** the devices, that is, we synchonize all the flip-flops to change states at the same time with a clocked pulse. Flip-flops are **edge triggered**; they either change states when the clock goes from 0 to 1 (positive/rising edge) or when the clock goes from 1 to 0 (negative/falling edge).

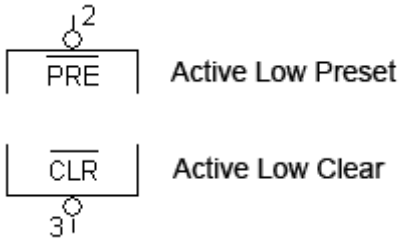
The symbols used for positive and negative edge triggering on flip-flops:



Asynchonous Inputs

Many flip-flops will also have a **clear** (CLR) and **preset** (PRE) terminal. These inputs are typically inverted, so they are active when the input signal is *low* (Active Low Input). The CLR and PRE signals can be asserted any time and don't have to be edge triggered; they will override any other inputs, including the clock.

The symbols used for *clear* and *preset* (the bubble indicates an inverted signal):



Truth Table for CLR and PRE (active low)

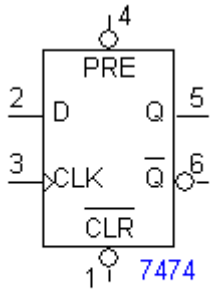
PRE	CLR	Output
0	0	Error state
0	1	1
1	0	0
1	1	Depends on inputs and clock

Types of flip-flops

There are several types of flip-flops but the two most important kind are the D and J-K flip-flops.

D flip-flop

Symbol for the D flip-flop:



Positive Edge Triggered
D Flip-Flop

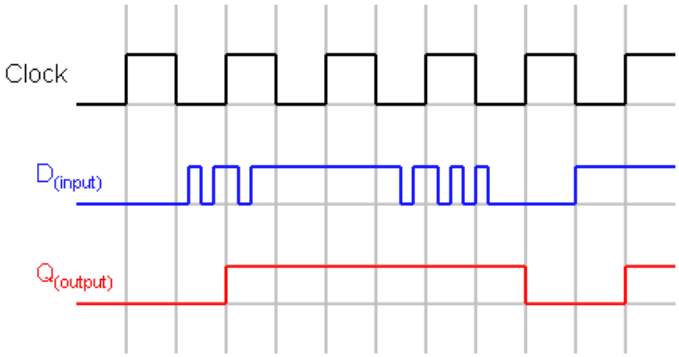
The D (Data) flip-flop has an input *D*, and the output *Q* will take on the value of *D* at every triggering edge of the clock pulse and hold it until the next triggering pulse. The D flip-flop is usually *positive edge triggered*.

The truth table for a positive edge triggered D flip-flop:

D	CK	Q
0	↑	0
1	↑	1
X	0,1	Q_0

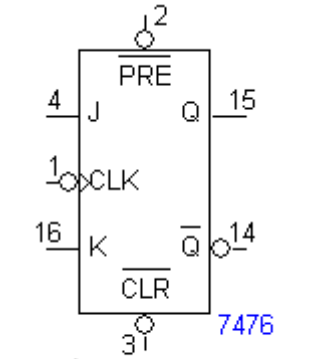
(↑ indicates a rising edge on the clock pulse; **X** indicates that it has no effect on outcome)

Timing diagram for the positive edge triggered D flip-flop:



JK flip-flop

Symbol for the JK flip-flop:



The JK flip-flop has two inputs, labeled *J* and *K*. *J* corresponds to a "set" signal, and *K* corresponds to a "reset" signal.

At the triggering edge:

If *J* is 1 and *K* is 0, *Q* is 1.

If *J* is 0 and *K* is 1, *Q* is 0.

If *J* and *K* are both 0, the output stays the same as it was before.

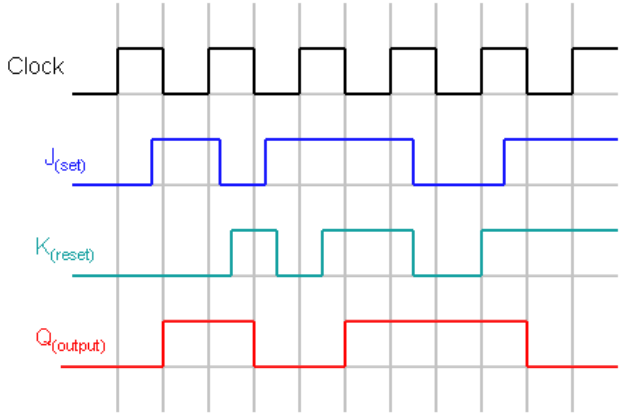
If *J* and *K* are both 1, the output is inverted.

The JK flip-flop is usually *negative edge triggered*.

The truth table for a negatively triggered JK flip-flop:

J	K	CK	Q
0	0	↓	Q_0
0	1	↓	0
1	0	↓	1
1	1	↓	Q'_0
X	X	0,1	Q_0

The timing diagram for the negatively triggered JK flip-flop:

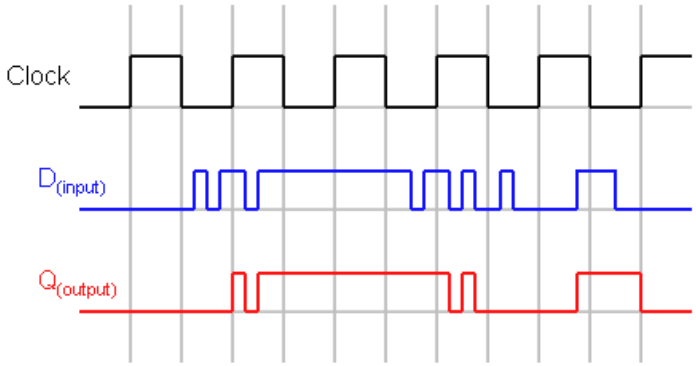


Latches

Latches are similar to flip-flops, but instead of being **edge triggered**, they are **level triggered**.

The most common type of latch is the **D latch**. While CK is high, Q will take whatever value *D* is at. When CK is low, *Q* will latch onto the last value it had before CK went low, and hold it until CK goes high again.

Timing diagram for the D latch:



Dva ključna elektronska sklopa digitalnog računara koji se grade iz sekvencijalnih elektronskih kola tipa flip-flopa i/ili leča su registar i brojač.

a1.6.2 Registri

Registar je elektronsko kolo koje služi za privremeno memorisanje, odnosno pamćenje podataka. Njegova primarna uloga je da prihvati delimične ili konačne rezultate nekog procesa obrade podataka, da ih zapamti i, kada je to potrebno, prosledi na dalju obradu.

Registar se može iskoristiti i u slučajevima kada je neophodno sinhronizovati dva digitalna kola koja rade različitim brzinama. Ovakav registar sa naziva bafer. Bafer svojom memorijskom funkcijom omogućava sinhronizaciju procesa koji su fizički asinhroni.

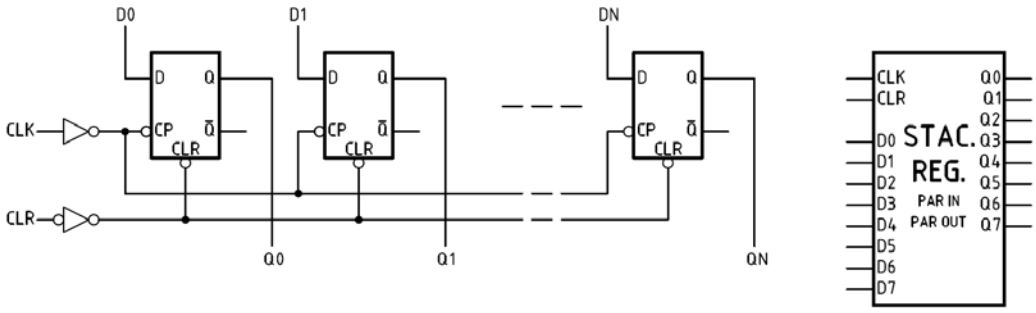
Osnovu za gradjenje registra čini flip flop. Kako flip flop može da memoriše samo jedan jednobitni podatak broj flip flopova mora da bude jednak memorijskom kapacitetu registra. Kapacitet registra je po pravilu mali i kreće se od nekoliko bitova do 100 kbita. Njihova osnovna odlika je velika brzina rada.

Ako se registar koristi samo da privremeno sačuva delimične ili konačne rezultate nekog procesa obrade, bez bilo kakvih drugih intervencija na unetom sadržaju, onda se takav registar naziva stacionarnim. Registar kod koga se vrši neka intervencija nad unetim sadržajem je dinamički registar.

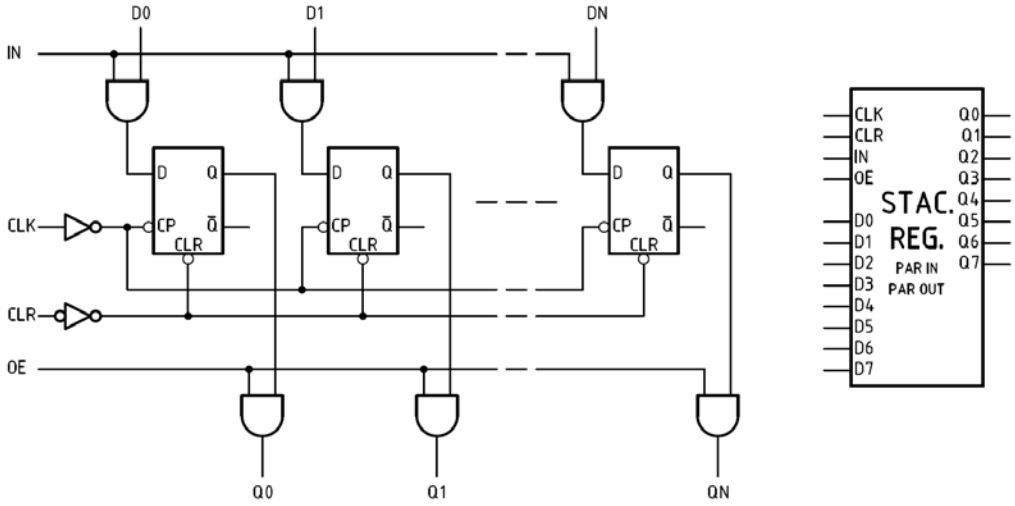
Stacionarni registar:

Stacionarni registri se u zavisnosti od pristupa kod upisa i čitanja dele na serijske i paralelne.

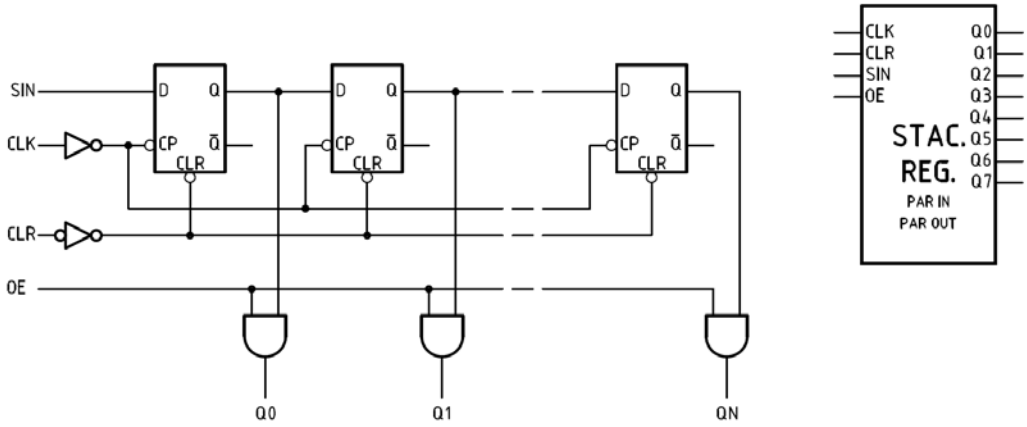
Primer registra sa paralelnim upisom i paralelnim čitanjem izvedenog pomoću D flip flopa:



Primer registra sa paralelnim upisom i paralelnim čitanjem sa dozvolom upisa i čitanja (upravljački ulazi IN i OE):



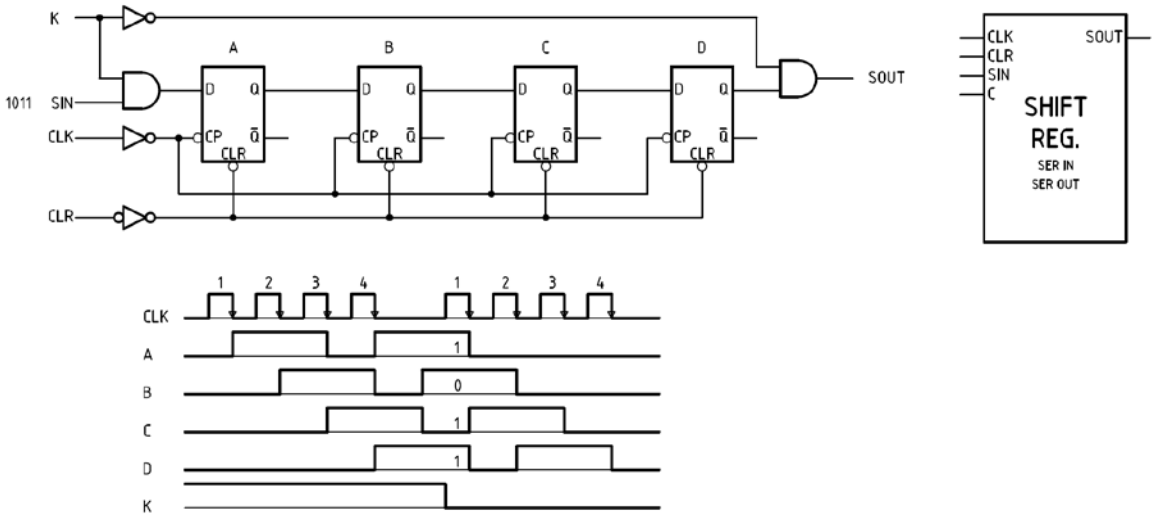
Primer registra sa serijskim upisom i paralelnim čitanjem sa dozvolom:



Dinamički registar:

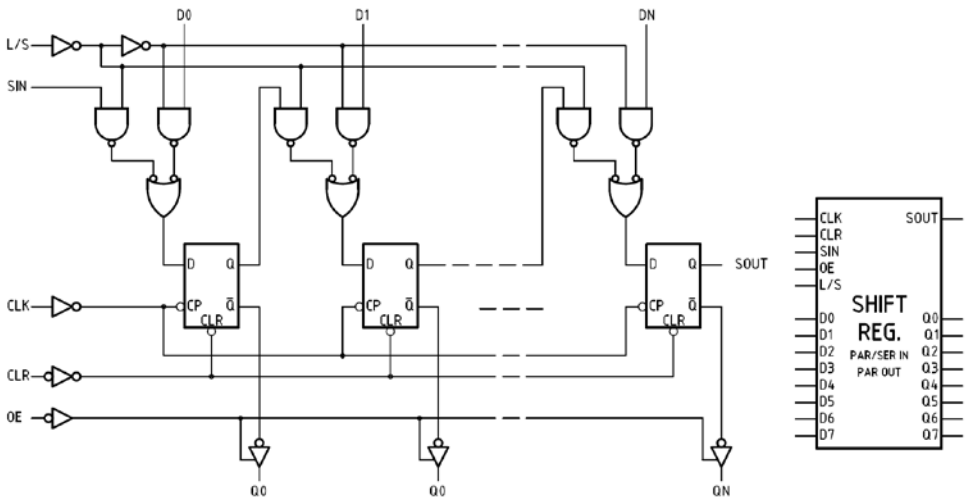
Dinamički registri su pomerački (shift) registri. Ukoliko se kod pomeranja bita obezbedjuje pravilo da se prvi upisani podatak prvi i čita, onda se radi o FIFO (First In First Out) pomeračkom registru. Kod registara kod kojih se poslednji upisani podatak prvi čita, radi se o LIFO (Last In First Out) pomeračkom registru.

Primer FIFO pomeračkog registra izvedenog pomoću D flip-flopa sa četiri bita i serijskim upisom i čitanjem:



Upravljački kanal K upravlja radom serijskog ulaza i izlaza. Kada je K=1 u registar se upisuje binarni sadržaj, što je uslovljeno I logičkim kolom na D ulaznom kanalu flip flopa A. Upis se ostvaruje za 4 takta. Taktni impulsi se dovode preko CLK kanala. Promena stanje D filpflopa nastaje na opadajućoj ivici taktnog signala CLK. Ukoliko se u trenutku pojave opadajuće ivice taktnog signala na ulazu, kanal SIN, nalazi logička jedinica, ona će biti upisana u flip flop A, u suprotnom stanje flip flopa A se ne menja. Istovremeno se stanje svih flip flopova (A, B, C i D) pri svakoj opadajućoj ivici taktnog signala prenosi na naredni flip flop, zato što je izlaz i-tog flip flopa povezan sa ulazom flip flopa (i+1). Posle 4 takta u registru se nalazi kompletna sekvenca dovedena na ulaz registra, ulazni kanal SIN. U konkretnom slučaju to je digit 1011. Sadržaj se čita na serijskom izlazu SOUT kada je upravljački signal K = 0. Kompletan sadržaj se prosledjuje na SOUT u četiri takta sukcesivnim premeštanjem sadržaja flip flopova kao kod upisa.

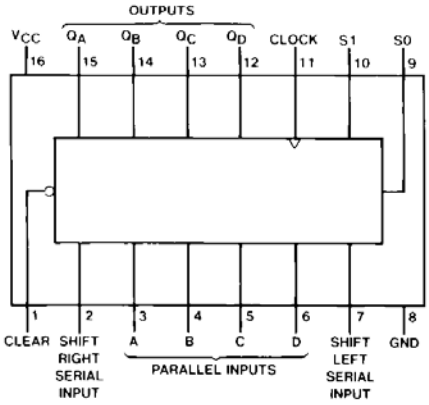
Primer univerzalnog pomeračkog registra sa paralelnim pisanjem i čitanjem:



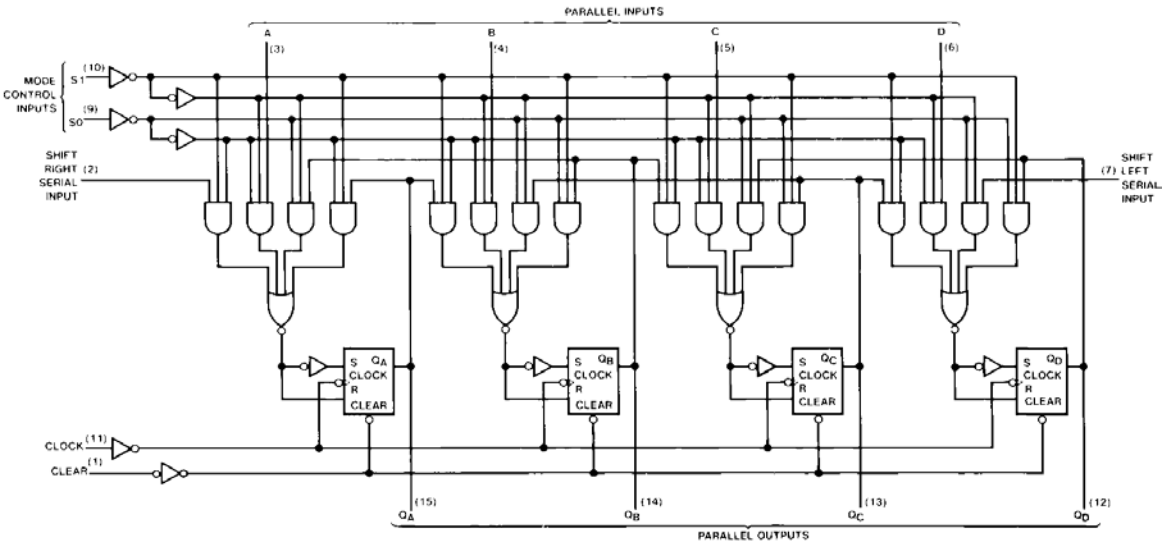
Kontrola paralelnog izlaza za čitanje registra ostvaruje se trostatičkim elementima.

4-bit Universal Shift Register 74LS194

Connection Diagram



Logic Diagram



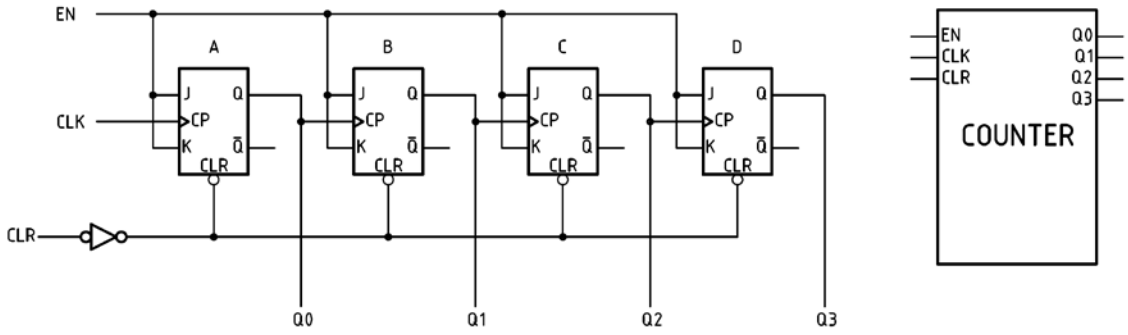
a1.6.2 Binarni brojač

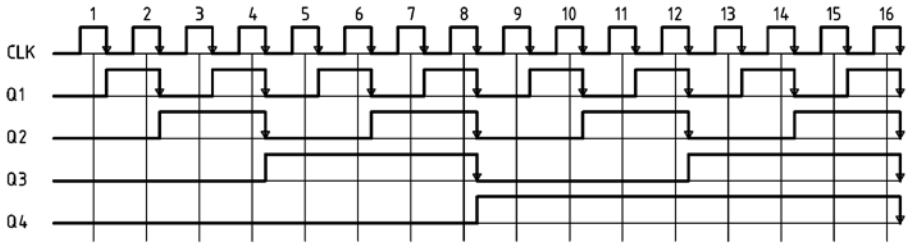
Brojači su sekvencijalne mreže. Broj različitih stanja se naziva modul ili osnova brojača. Brojač sa m stanja je brojač modula m.

Flip flop se može direktno primeniti za sintezu binarne mreže koja može da broji impulse na svom ulaznom kanalu. Brojač od n flip flopova se naziva n-bitni binarni brojač, ili binarni brojač modula 2^n. Ako se svi flip-flopovi u brojaču taktuju zajedničkim taktnim impulsom, takvi brojači se nazivaju sinhronim, u suprotnom, oni su ansinhroni.

Moguće su tri osnovne varijante brojačkih mreža: brojačka mreža za brojanje unapred, brojačka mreža za brojanje unazad i reverzibilna brojačka mreža koji ostvaruje brojanje u oba smera.

Asinhroni binarni brojač: Primer četvorobitnog brojača sa JK flip-flopom. Na ulaz FFO flip flopa dovodi se taktni signal ili aperiodični binarni signal koje generiše neki drugi spoljašnji uredjaj čije se promene stanja broje. Modul ovog brojača je 2^4 = 16. Pažljivim posmatranjem taktnog dijagrama može se uočiti da je frekvencija svakog narednog flip flopa dva puta manja od prethodnog, uključujući i generator takta. Ova činjenica se koristi kod izgradnje modula za deljenje frekvencije, što se vrlo često sreće kod digitalnih sistema.



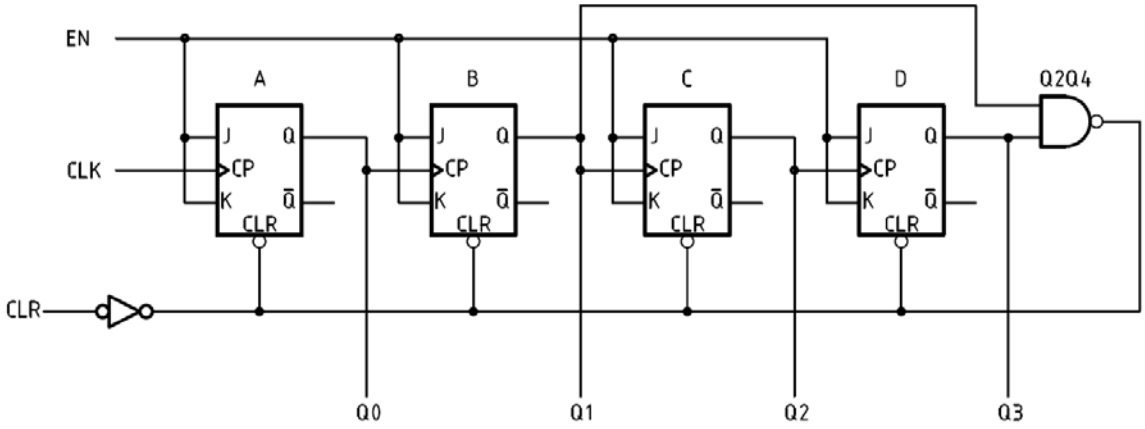


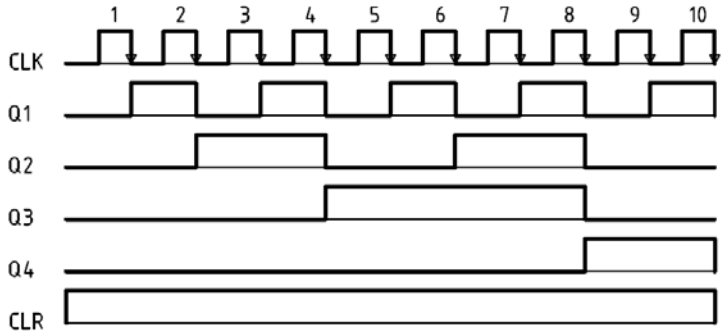
Takt	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18
Q0	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1
Q1	0	0	1	1	0	0	1	1	0	0	1	1	0	0	1	1	0	0
Q2	0	0	0	0	1	1	1	1	0	0	0	0	1	1	1	1	0	0
Q3	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1	0	0
DE	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	0	1

Dovodjenjem Q3:Q0 na BCD/7Seg konvertor koda i zatim na sedmosegmetni displej, moguće je pokazati kako funkcioniše ovaj brojač. Da bi se izbegla kodne kombinacije koje ne odgovaraju ciframa decimalnog brojnog sistema, onda se modul četvorobitnog brojača treba da ograniči na 10. Time se dobija takozvani brojač proizvoljnog modula brojanja.

Brojač proizvoljnog modula brojanja:

Primenom JK flip flopova sa reset kanalom, moguće je ugradnjom dodatnog logičkog kola resetovati brojač pre nego što on dostigne svoj binarni kapacitet, odnosno binarni modul. U konkretnom primeru, prikazano je kako se od četvorobitnog brojača formira dekadni brojač. U trenutku kada brojač generiše unutrašnje stanje koje odgovara binarno kodiranoj cifri $10_{10} = 1010_2$ brojač se resetuje i dovodi u svoje inicijalno stanje, odnosno na decimalnu cifru nula. Reset funkcija se jednostavno ostvaruje I kolom koje sumira stanje izlaza na flip flopu B i flip flopu D.





Takt	1	2	3	4	5	6	7	8	9	10	11	12	13	...
Q0	0	1	0	1	0	1	0	1	0	1	0	1	0	
Q1	0	0	1	1	0	0	1	1	0	0	0 1	0	1	
Q2	0	0	0	0	1	1	1	1	0	0	0	0	0	
Q3	0	0	0	0	0	0	0	0	1	1	0 1	0	0	
DE	0	1	2	3	4	5	6	7	8	9	0	1	2	

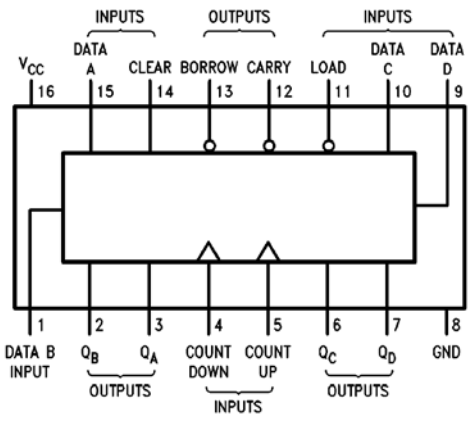
Primer sinhronog četvorobitnog dvosmernog brojača: 74LS193

54LS193/DM54LS193/DM74LS193 Synchronous 4-Bit Up/Down Binary Counters with Dual Clock

This circuit is a synchronous up/down 4-bit binary counter. Synchronous operation is provided by having all flip-flops clocked simultaneously, so that the outputs change together when so instructed by the steering logic. This mode of operation eliminates the output counting spikes normally associated with asynchronous (ripple-clock) counters.
The outputs of the four master-slave flip-flops are triggered by a low-to-high level transition of either count (clock) input. The direction of counting is determined by which count input is pulsed while the other count input is held high.
The counter is fully programmable; that is, each output may be preset to either level by entering the desired data at the inputs while the load input is low. The output will change independently of the count pulses. This feature allows the counters to be used as modulo-N dividers by simply modifying the count length with the preset inputs.
A clear input has been provided which, when taken to a high level, forces all outputs to the low level; independent of the count and load inputs. The clear, count, and load inputs are buffered to lower the drive requirements of clock drivers, etc., required for long words.These counters were designed to be cascaded without the need for external circuitry. Both borrow and carry outputs are available to cascade both the up and down counting functions. The borrow output produces a pulse equal in width to the count down input when the counter underflows.
Similarly, the carry output produces a pulse equal in width to the count down input when an overflow condition exists. The counters can then be easily cascaded by feeding the borrow and carry outputs to the count down and count up inputs respectively of the succeeding counter.

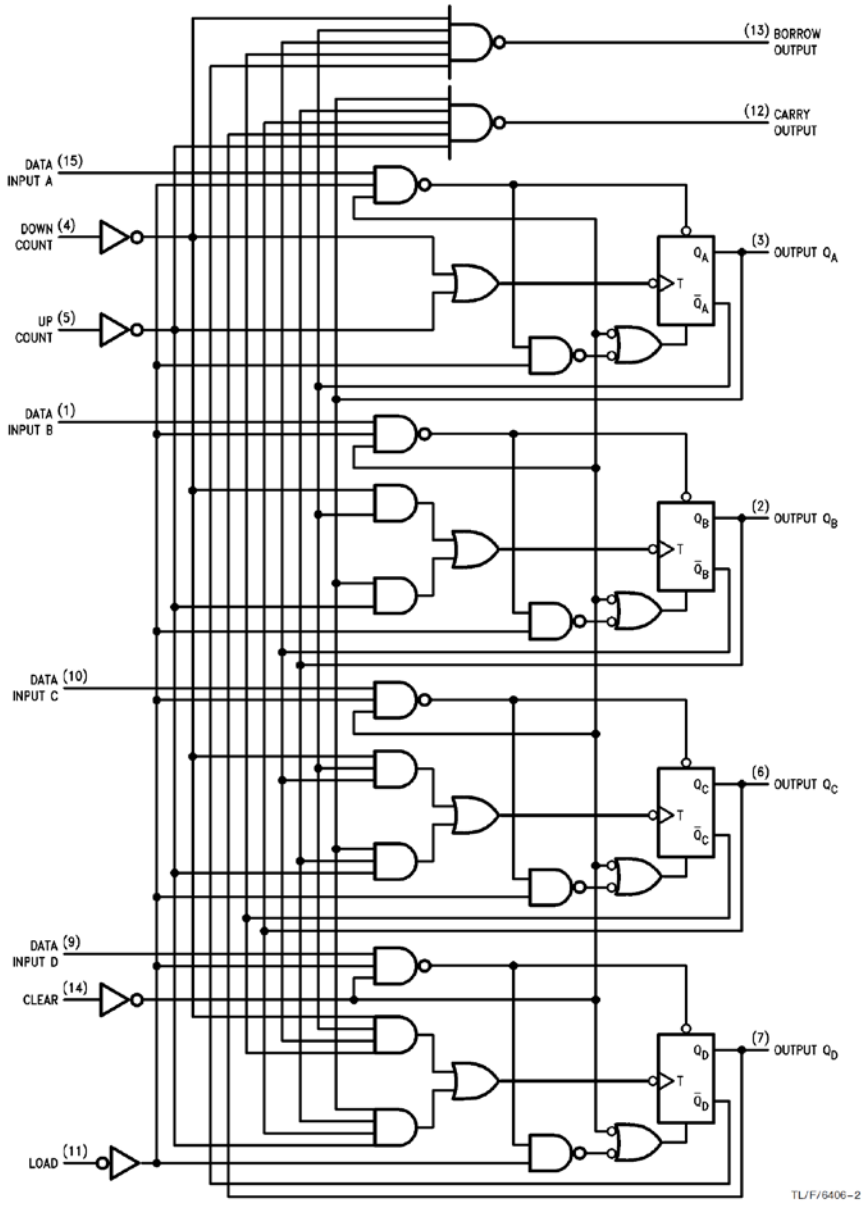
Connection Diagram

Dual-In-Line Package



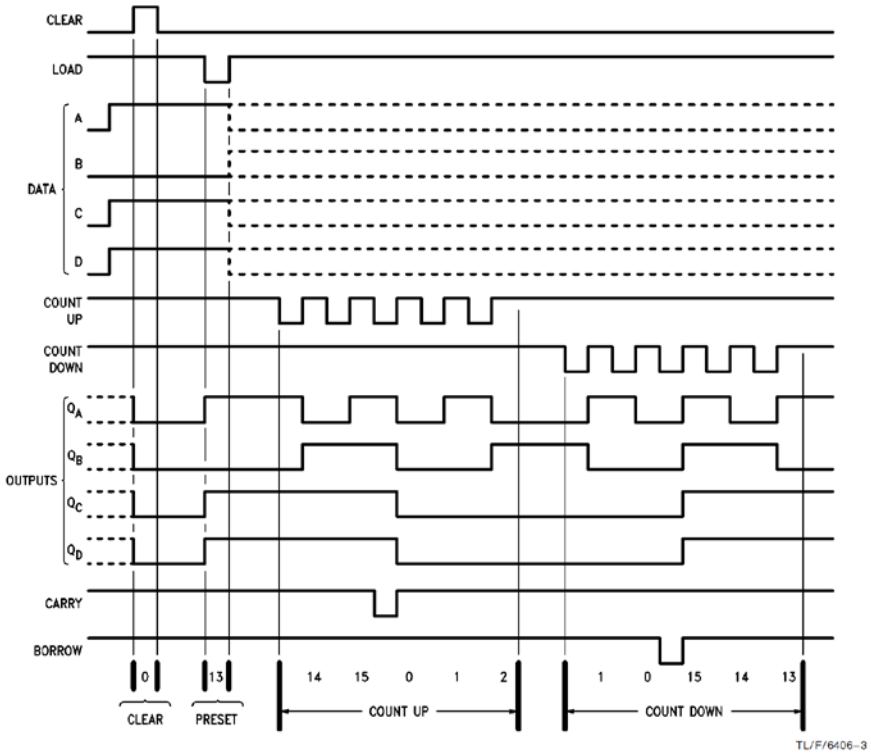
TL/F/6406–1

Logic Diagram



Timing Diagrams

Typical Clear, Load, and Count Sequences

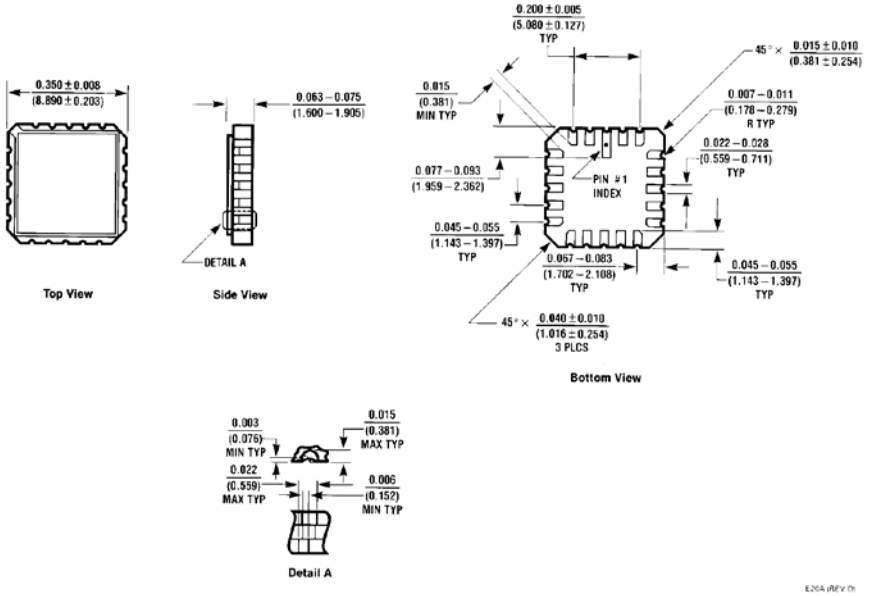


TL/F/6406–3

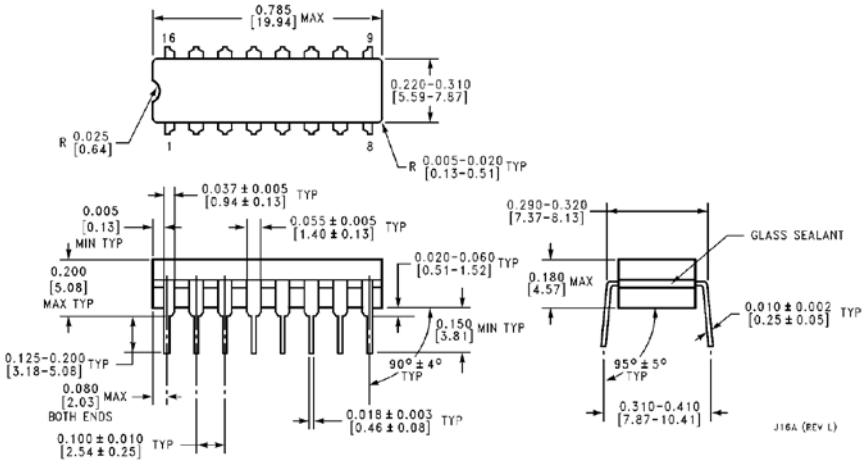
Note A: Clear overrides load, data, and count inputs.

Note B: When counting up, count-down input must be high; when counting down, count-up input must be high.

Physical Dimensions inches (millimeters)



Ceramic Leadless Chip Carrier Package (E)
Order Number 54LS193LMQB
NS Package Number E20A



16-Lead Ceramic Dual-In-Line Package (J)
Order Number 54LS193DMQB or DM54LS193J
NS Package Number J16A

a1.6.2 Časovnik

Dovodjenjem na ulaz signala koji generiše neki uniformni taktni generator pravougaonog impulsa (astabilni multivibrator konstantne frekvencije ili kristalni oscilator), brojač se pretvara u časovnik sa vremenskom bazom jednakoj periodi oscilovanja taktnog generatora.

Promena vremenske baze ostvaruje se primenom delitelja frekvencije. Delitelj frekvencije je elektronski sklop koji inkrementalno povećava vremensku bazu po geometrijskoj progresiji sa osnovom 2 (2, 4, 8, 16, ...). Sklop asinhronog binarnog brojača je vrlo pogodan za ovu funkciju. Uzimanjem signala sa Q_0 izlaza, dobija se dva puta manja frekvencija od frekvencije pobudnog taktnog generatora. Na izlazu Q_1 generiše se četiri puta manja frekvencija, a na izlazu Q_2, osam puta manja frekvencija i tako dalje.

Kaskadnom vezom brojačkih modula časovnika moguće je napraviti složene sisteme za merenje vremena, kao što je to kalendarski časovnik, koji meri sekunde, minute, sate, dane i dalje složenije vremenske kvante kao što su meseci i godine. U ovom slučaju se koristi koncept časovnika proizvoljnog modula brojanja, pri čemu se na primer, časovnik koji broji minute projektuje kao brojač za heksagezimalnim modulom, odnosno modulom 60, i tako redom.

MAKING A WORLD OF DIFFERENCE:

Engineering Ideas into Reality

1964
Dawn of the Digital Age

By the mid-1960s, thanks to the work of engineers in the decades just before and after World War II, Americans were accustomed to many conveniences in daily life. Tap water was safe to drink. Electric power was reliable and affordable. And the family could take its summer road trip on the new interstate highway system—including bridges, tunnels, rest stops for gas and food, and standardized signage—that connected an ever-growing number of cities and towns from coast to coast.

Today these conveniences are so commonplace that we think about them only when there's a problem—the power is out, there's a water main break, or two lanes on a bridge are closed for repairs. But when they first occurred, these advances and innovations had a profoundly transformative effect on the nation and on individuals and families. To cite just a couple of statistics: by the 1930s, the creation of sewage sanitation systems and public supplies of clean drinking water had virtually eliminated the spread of waterborne diseases like cholera and typhoid. Combined with other public health advances such as vaccination programs, antibiotics, and a safer food supply, those crucial improvements in sanitation and water supply helped increase the average life expectancy in the United States by 50 percent—from 47 years to 70 between 1900 and 1960. (By comparison, average life expectancy since the mid-1960s has increased only about 12 percent.) Similarly, by the 1940s, a few years after the establishment of the Rural Electrification Administration in 1935, 800 rural electric cooperatives had been formed and 350,000 miles of power lines brought millions of Americans in rural communities literally out of the "dark ages."

None of those advances happened by chance. In response to public demand, public policy, and an intrinsic creative drive, engineers created the infrastructure essential to the health, prosperity, and security of the American people—not only for electrification, sanitation, and water supply and distribution, but also for automobiles, highways, refrigeration, air-conditioning, aviation, high-performance materials, and much, much more. Nor did progress stop there. As would become clear in the latter part of the 20th century, engineering innovations between the mid-1940s and mid-1960s, many driven by Cold War national security concerns and the Department of Defense, were quietly laying the foundation for scores of advances that Americans in the 21st century would take for granted—advances in computers, communications, and health care, among other fields.

In 1964, glimmers of the changes that would transform American society were beginning to enter public awareness. Even as nuclear arms deployed by the United States and the Soviet Union in the ongoing Cold War loomed large, peaceful uses of atomic energy were emerging. The first nuclear power plants in the United States came online in the late 1950s; the use of nuclear medicine procedures for diagnostics and treatment, which had begun in the 1930s, expanded in the 1960s. Another recent invention, the laser, would soon demonstrate its value to health care and fiber-optic communications. And while the Air Force was using room-size mainframe computers to process data from far-flung radar stations and guard against attacks by Soviet bombers, the introduction of much smaller and more versatile computing machines was about to alter life in the United States and the world at large forever. A new era was dawning—a digital age that would transform how we lived, worked, and communicated.

ENIAC *(right)* ran on nearly 18,000 vacuum tubes and needed a staff to plug in thousands of wires to set or change its program. With the advent of the transistor *(far right)*, vacuum tubes became obsolete and computers began to shrink.

Electronics: Smaller, Faster, Cheaper

In the late 1940s and 1950s, electronic computers were still enormous and enormously expensive. They were the province of large institutions—governments, big corporations, universities, and especially the military—that could afford to buy them, build cooled rooms large enough to house them, and hire the operators to make them work. In 1946, a behemoth named ENIAC (Electronic Numerical Integrator and Computer) was unveiled at the University of Pennsylvania. It weighed 30 tons, occupied a room 30 feet by 50 feet, and operated with nearly 18,000 bulky, power-hungry vacuum tubes that frequently burned out. Commissioned to produce artillery firing tables so gunners in the field could adjust their aim as needed, ENIAC could perform in just 30 seconds calculations that used to take 12 hours on a hand calculator.

Powerful as it was, ENIAC had limitations. For one thing, this computer had only enough memory to handle the numbers involved in the current computation; its instructions, or program, had to be wired into the circuitry. So, changing the program meant someone had to spend several days unplugging and replugging thousands of wires to enter the changes and then test the new settings.

Even as ENIAC was coming online, engineers elsewhere were exploring a different way to build these machines. In 1947, a revolutionary engineering advance emerged from AT&T Bell Labs that would start society on the path to the current era of both hyper-fast supercomputers and the ubiquitous smartphone.

This innovation—a device based on solid-state semiconductor materials that could both amplify an electrical signal and turn it on and off—was the result of a brilliant collaboration among John Bardeen, William B. Shockley, and Walter H. Brattain. The team had been asked to develop a replacement for vacuum tubes, which were not only unreliable power hogs but also could not pick up the ultrahighfrequency radio waves needed for AT&T's transcontinental telephone system. Two days before Christmas 1947, after a month of intense experimentation, the team presented their bosses at Bell Labs with the transistor *(above)*.

The transistor generated very little heat and was both dependable and tiny—characteristics that would lead to a phenomenal miniaturization of complex circuitry, paving the way for virtually every electronic device we rely on today. For their monumental "researches on semiconductors and their discovery of the transistor effect," Bardeen, Shockley, and Brattain shared the Nobel Prize in Physics in 1956.

Within a few years, engineers were using transistors to produce small devices that amplified sound, such as transistorized hearing aids and pocket-size transistor radios. By the mid-1960s, as transistor design and manufacturing improved, computer engineers used them to

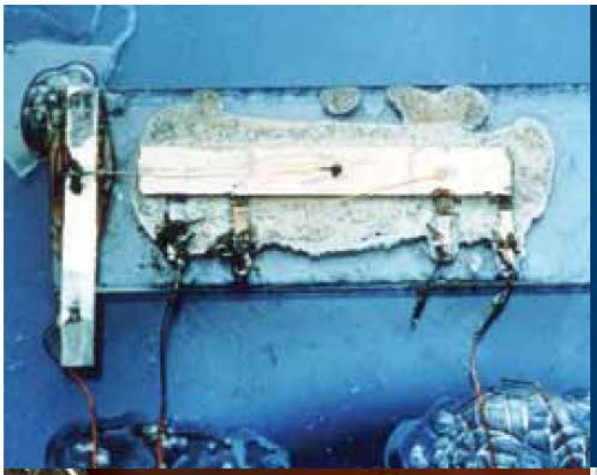
build a new generation of supercomputers, like Control Data Corporation's CDC 6600. Designed by Seymour Cray, the CDC 6600 was almost three times faster than the next fastest machine of its day, the IBM 7030 Stretch. Despite being phenomenally fast and much more reliable and efficient than ENIAC, the CDC 6600 was still a huge machine with a huge price tag. At $7 to $10 million apiece, it was not something your average business, and certainly not your average consumer, could afford.

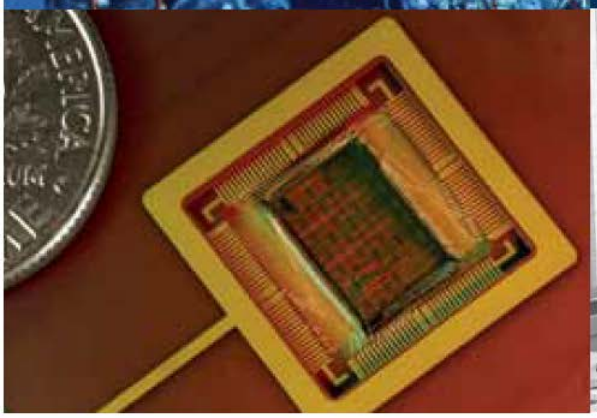
The crucial engineering advance that brought computers out of large institutions and into much wider use was the integrated circuit, developed independently in the late 1950s by Jack Kilby of Texas Instruments and Robert Noyce of Fairchild Semiconductor, a pioneering firm in California's Silicon Valley. In 1989, Kilby and Noyce

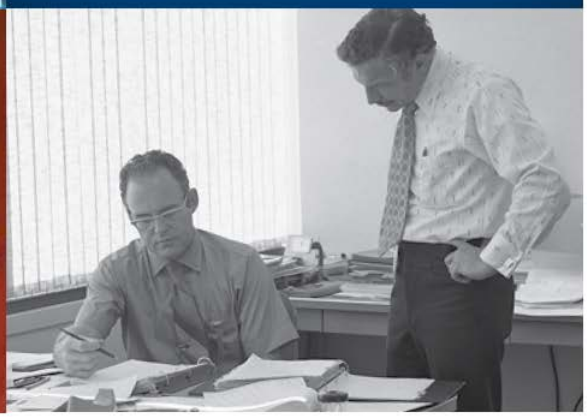
would be awarded the first Charles Stark Draper Prize for Engineering, the National Academy of Engineering's (NAE's) highest award, for "their independent co-invention of the monolithic (meaning formed from a single crystal) integrated circuit, better known as the semiconductor microchip." Robert Noyce died in 1990, but in 2000 Jack Kilby was awarded half of that year's Nobel Prize in Physics "for his part in the invention of the integrated circuit." (The other half of the prize was shared by Zhores I. Alferov and Herbert Kroemer "for developing semiconductor heterostructures for high-speed- and opto-electronics.") The integrated circuit squeezed multiple transistors, wiring, and other components of an electronic circuit onto a single silicon chip using photographic techniques to reduce the circuit design to a tiny imprint, which was then printed on a wafer the size of a baby's fingernail.

Integrated circuits produced in the 1960s were essential to early aerospace projects such as the Minuteman missile and the Apollo program, which both needed lightweight digital computers for their inertial guidance systems. This early government support allowed integratedcircuit makers to refine manufacturing methods and lower costs enough to enter the industrial and, eventually, the consumer markets.

As production costs came down, the average price per integrated circuit dropped from $50 in 1962 to $2.33 in 1968, even as the number of transistors on a chip skyrocketed. In 1965 Gordon Moore—who worked with Noyce at Fairchild Semiconductor and later joined him as cofounder of Intel Corporation—predicted that computing capacity, based on the number of transistors packed into a chip, would double every year. The race toward ever smaller yet ever more powerful computers was off and running. (Updating his forecast in 1975, Moore predicted that chip capacity would double every two years, an estimate that remained close to the mark for decades. "Moore's Law" is still used today as a standard for measuring industry progress—a testament to the creativity and ingenuity of engineers focused on improving both performance and cost.) Computers as we know them would not exist, of course, without the ingenuity of the programmers and software engineers who created the programming languages, operating systems, and applications that make the machines useful in so many different ways. High-level programming languages like Fortran, COBOL, and BASIC were instrumental in making programming faster and considerably less tedious than hand-coding in the ones and zeros of machine language.







From a few transistors in the first integrated circuit *(left)*, the number of components crammed on a microchip doubled every two years, as predicted by Gordon Moore (shown seated, *below*, with Robert Noyce, one of the inventors of the integrated circuit). In 1995, University of Pennsylvania engineering students designed "ENIAC on a chip"—recreating the 30-ton ENIAC's circuits with 250,000 transistors on a chip only 8 mm square *(below, left)*.

FORTRAN

(FORmula TRANslating System) was developed in the mid-1950s by an IBM team led by John Backus. "Much of my work has come from being lazy," Backus told *Think,* the IBM employee magazine, in 1979. "I didn't like writing programs, and so, when I was working on the IBM 701 [an early computer], writing programs for computing missile trajectories, I started work on a programming system to make it easier to write programs." Designed for scientific and engineering applications, some version of Fortran is still used in intensive supercomputing tasks such as weather and climate modeling, computational fluid dynamics, and structural engineering. John Backus was awarded the NAE's Draper Prize in 1993 for "development of FORTRAN, the first widely used, general purpose, high-level computer language."

COBOL

(COmmon Business-Oriented Language) was created by a committee of computer manufacturers and their clients, notably the government. A key member of the committee was the indomitable programmer Rear Admiral Grace Murray Hopper, who had long believed that programming languages should be usable by people who were neither mathematicians nor computer experts. The goal was to create a language suited to large-scale data processing such as for payrolls, budgets, and inventory—and to have programs that could run on different makes of machines. This compatibility was especially important to the Department of Defense (DOD), which bought computers from different manufacturers. In December 1960, the same COBOL program ran successfully on both a Remington Rand UNIVAC II and an RCA 501. COBOL would dominate government and business data processing for decades and is still used for millions of banking transactions today.

BASIC

(Beginner's All-purpose Symbolic Instruction Code) was invented in 1963 at Dartmouth College by mathematicians John Kemeny and Thomas Kurtz *(below)* as a teaching tool for undergraduates. Kemeny and Kurtz had the radical idea that undergrads—science and nonscience majors alike—could learn about computing by actually writing their own programs. But first they needed a more user-friendly language. The language they created used simple English words such as PRINT, SAVE, and RUN. To get the computer to write something you merely typed PRINT, followed by the words to print in quotes. Kemeny wanted the language to be so easy that a complete novice "could use it after three hours of training." Versions of BASIC became popular with the advent of minicomputers such as Digital Equipment Corporation's PDP line in the mid-1960s and then exploded with the introduction of home computers in the mid-1970s. (Bill Gates and Paul Allen wrote a version of BASIC for the MITS [Micro Instrumentation Telemetry System] Altair and then went on to form Microsoft—and the rest, as they say, is history.)

THE MINICOMPUTER

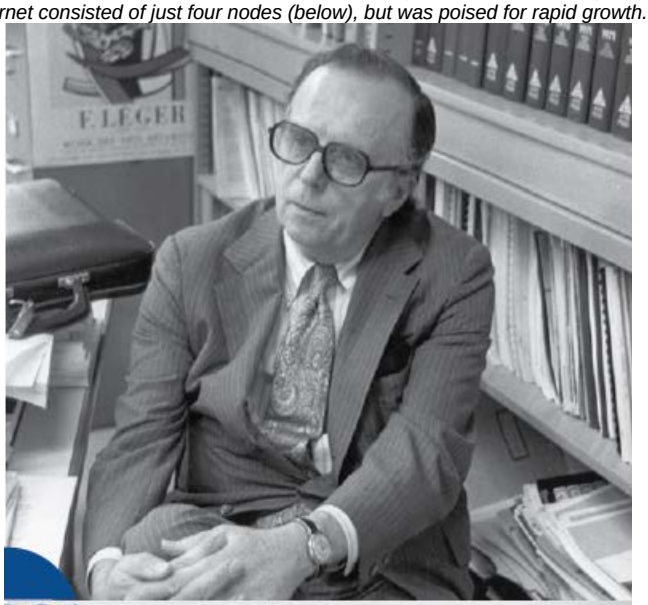
Affordable, Compact, and User-Friendly

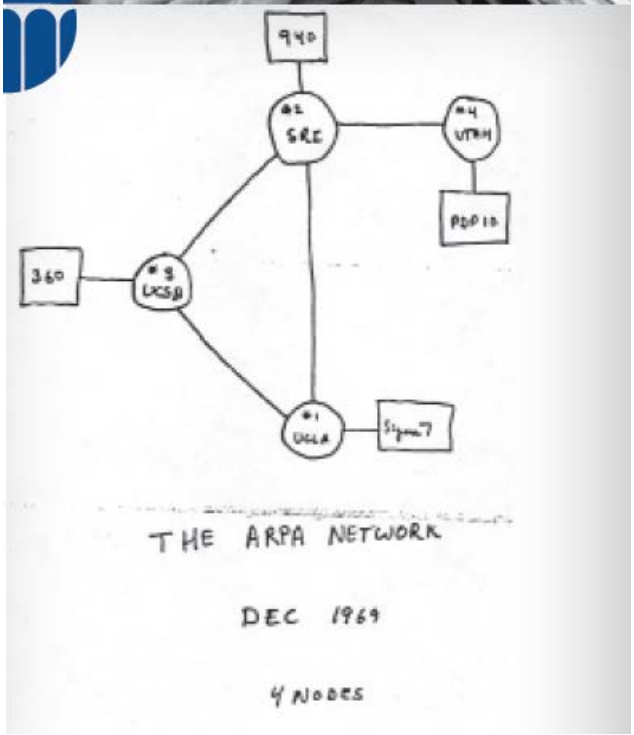
In 1965, Digital Equipment Company introduced the PDP-8—eighth in a revolutionary line of interactive computers that focused on the user's experience rather than solely on machine efficiency. Sold for $18,000 and available in a desktop configuration, the PDP-8 was the first commercially successful minicomputer, affordable for many midsize businesses and small laboratories. A Digital executive in England, where small cars and short skirts were in fashion in the 1960s, was credited with coining that term in a sales report: "Here is the latest minicomputer activity in the land of miniskirts as I drive around in my [Austin] Mini Minor." To promote the machine's small size, the company photographed it in the back of a Volkswagen Beetle *(above)*. Soon the PDP-8 was at work in many settings, from controlling the baseball scoreboard at Boston's Fenway Park *(opposite, top)* and the lights at a New York theater to doing signal analysis in physics labs and monitoring instruments in a hospital operating room *(opposite, bottom)*. In 1970, the PDP-8/E came along, priced at only $6,500, with a configuration that allowed devices such as teletypewriters and line

Building the Digital Highway

Interdependence, connection, and collaboration were values that drove the engineers who created the Internet. J. C. R. Licklider, the visionary first director of the Information Processing Techniques Office (IPTO) at DOD's Advanced Research Projects Agency (ARPA) in the early 1960s, was chief evangelist for a radical new idea: connecting computers in such a way that all users could have access to data and software from anywhere. During his tenure at IPTO, Licklider funded research for three seminal developments in information technology—creation of computer science and engineering departments at several major universities, time-sharing, and networking. His ideas and the work of the many people he sponsored led, directly or indirectly, to the interconnected information age we live in today.

J. C. R. ("Lick") Licklider envisioned a network of connected computers that gave users access to programs and data anywhere. In 1969, that ancestor to the Internet consisted of just four nodes (below), but was poised for rapid growth.

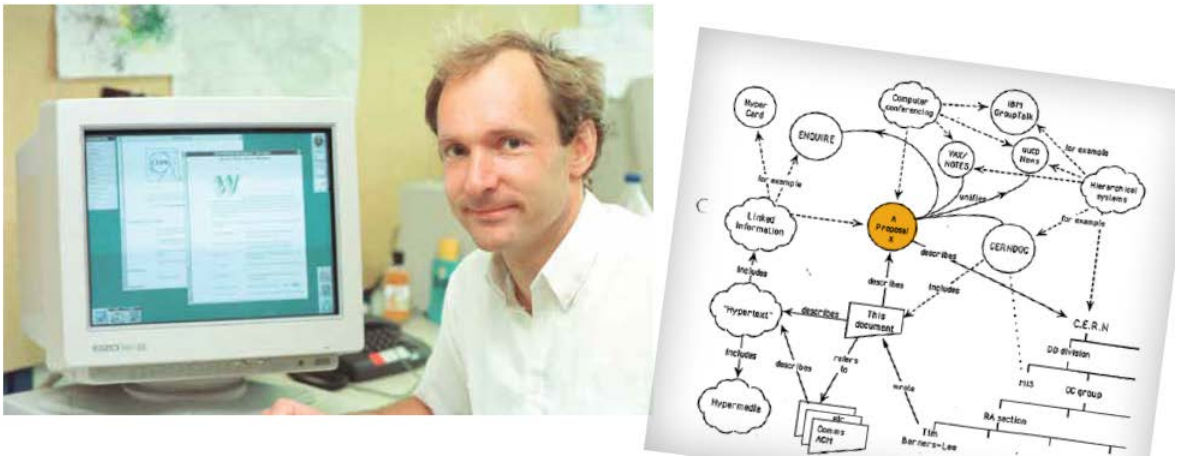




....

Birth of the World Wide Web

e-mail was both an essential tool in the collaborative work needed to create the Internet in the first place and a new model for person-to-person communication. But for Tim Berners-Lee, a software engineer at CERN, the European Organization for Nuclear Research in Switzerland, e-mail was not sufficient.



Tim Berners-Lee, shown at CERN with the NeXT computer he used to invent the World Wide Web, wrote his revolutionary proposal for the Web in March 1989. The cover of the proposal (above) sketches how hypertext links would allow users to follow their interests from source to source.

In 1989, Berners-Lee shared the frustration of many of his colleagues at the difficulty of keeping track of experiments and information in their fast-paced world. In his observation, people responding to an article posted by one scientist might refer not just to that message or topic but also to each other's messages or topics, creating a dense web of digital information in which researchers found it increasingly difficult to locate material relevant to their own research. The best way to access and share that store of knowledge, Berners-Lee concluded, was to use the technique known as hypertext, with links that let a reader jump from the mention of a document to the document itself, allowing users to navigate CERN's huge store of information in any direction.

That March, Berners-Lee submitted a plan for "information management" to his boss at CERN, who called it "vague but interesting." Given the go-ahead to flesh out the proposal, Berners-Lee and Belgian systems engineer Robert Cailliau grafted the hypertext idea onto the Transfer Control Protocol (TCP) and Domain Name System (DNS) already in use on the Internet. The resulting Hypertext Transfer Protocol—HTTP—formed the basis for what would become the World Wide Web, which they described as a web of hypertext documents that "browsers" could view.

In December 1990, they demonstrated prototype software for a basic Web system at CERN. Each file was tagged with the prefix "http," followed by "www" (World Wide Web) and a uniform resource locator (URL) identifying the site's physical host along with the name and location of the file in the host's directory. Visitors to the first Web page—at CERN—could learn about hypertext and the Web project itself, as well as find technical details for creating their own Web pages.

.....