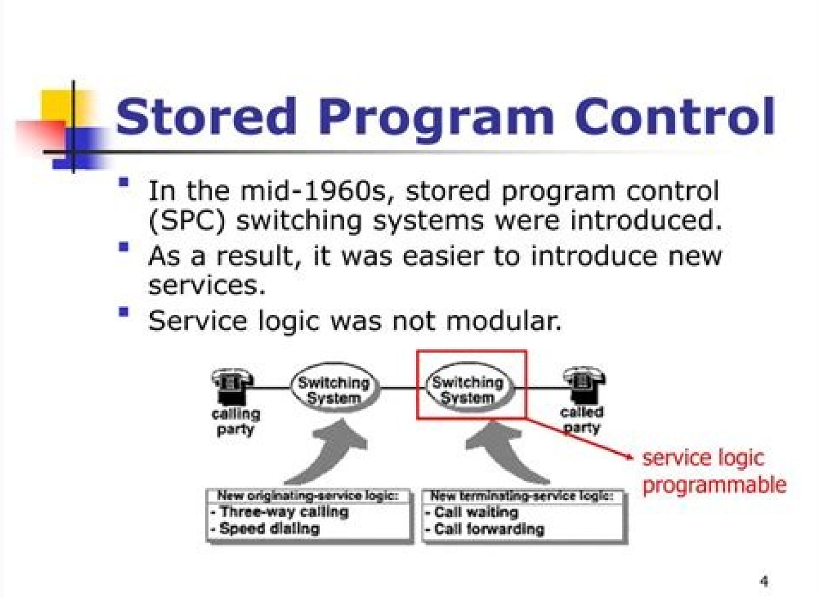
Stored program control electronic switching

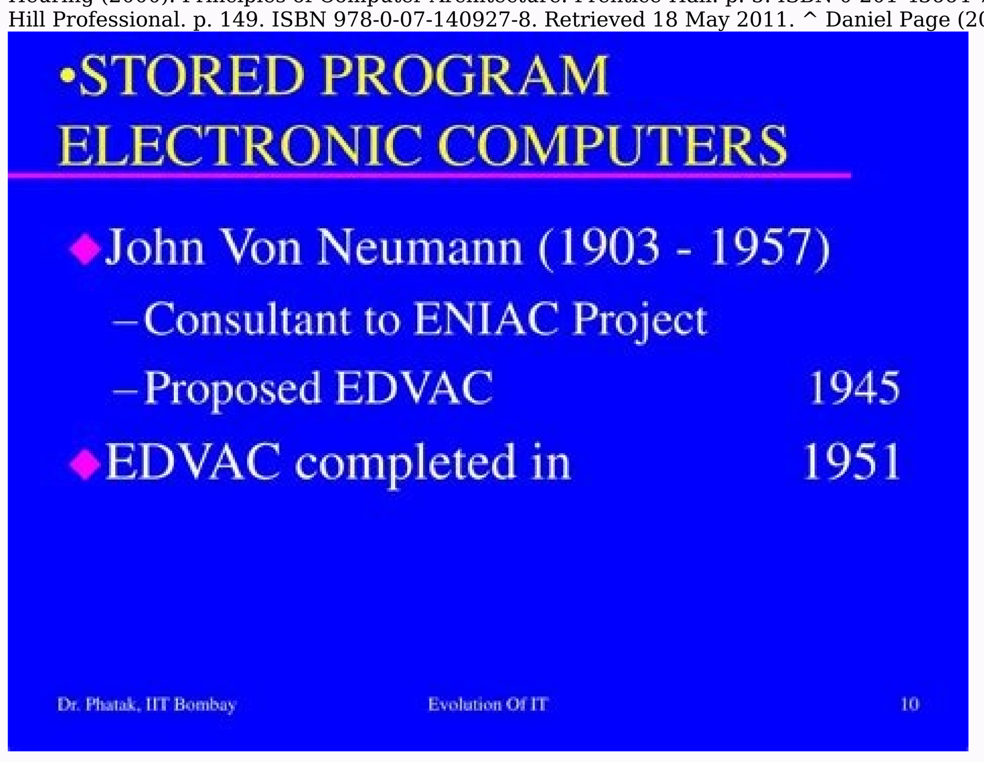
Stored program controlled.

See also: Universal Turing machine: Stored-program computer Computer that stores program instructions in electronically or optically accessible memory A stored-program computer is a computer that stores program instructions in electronically or optically accessible memory.[1] This contrasts with systems that stored the program instructions with plugboards or similar mechanisms. The definition is often extended with the requirement that the treatment of programs and data in memory be interchangeable or uniform.[2][3][4] Description In principle, stored-program computers have been designed with various architectural characteristics. A computer with a von Neumann architecture stores program data and instruction data in the same memory, while a computer with a Harvard architecture has separate memories for storing program and data.[5][6] However, the term stored-program computer is sometimes used as a synonym for the von Neumann architecture.[7][8] Jack Copeland considers that it is "historically inappropriate, to refer to electronic stored-program digital computers as 'von Neumann machines'".[9] Hennessy and Patterson wrote that the early Harvard machines were regarded as "reactionary by the advocates of stored-program computers".[10] History The concept of the stored-program computer can be traced back to the 1936 theoretical concept of a universal Turing machine.[11] Von Neumann was aware of this paper, and he impressed it on his collaborators.[12] Many early computers, such as the Atanasoff–Berry computer, were not reprogrammable. They executed a single hardwired program. As there were no program instructions, no program storage was necessary. Other computers, though programmable, stored their programs on punched tape, which was physically fed into the system as needed. In 1936, Konrad Zuse anticipated in two patent applications that machine instructions could be stored in the same storage used for data.[13][14] The University of Manchester's Baby[15] is generally recognized as world's first electronic computer that ran a stored program—an event that occurred on 21 June 1948.[16][17] However the Baby was not regarded as a full-fledged computer, but more a proof of concept predecessor to the Manchester Mark 1 computer, which was first put to research work in April 1949. On 6 May 1949 the EDSAC in Cambridge ran its first program, making it another electronic digital stored-program computer.[18] It is sometimes claimed that the IBM SSEC, operational in January 1948, was the first stored-program computer;[19] this claim is controversial, not least because of the hierarchical memory system of the SSEC, and because some aspects of its operations, like access to relays or tape drives, were determined by plugging.[20] The first stored-program computer to be built in continental Europe was the MESM, completed in the Soviet Union in 1950.[21] The first stored-program computers Several computers could be considered the first stored-program computer, depending on the criteria.[3] IBM SSEC, became operational in January 1948 but was electromechanical[22] In April 1948, modifications were completed to ENIAC to function as a stored-program computer, with the program stored by setting dials in its function tables, which could store 3,600 decimal digits for instructions. It ran its first stored program on 12 April 1948 and its first production program on 17 April[23][24] ARC2, a relay machine developed by Andrew Booth and Kathleen Booth at Birkbeck, University of London, officially came online on 12 May 1948.[25] It featured the first rotating drum storage device.[26][27] Manchester Baby, a developmental, fully electronic computer that successfully ran a stored program on 21 June 1948. It was subsequently developed into the Manchester Mark 1, which ran its first program in early April 1949. Electronic Delay Storage Automatic Calculator, EDSAC, which ran its first programs on 6 May 1949, and became a full-scale operational computer. EDVAC, conceived in June 1945 in First Draft of a Report on the EDVAC, but not delivered until August 1949. BINAC, delivered to a customer on 22 August 1949. It worked at the factory but there is disagreement about whether or not it worked satisfactorily after being delivered. If it had been finished at the projected time, it would have been the first stored-program computer in the world. It was the first stored-program computer in the U.S.[28] Manchester University Transistor Computer, is generally regarded as the first transistor-based stored-program computer having become operational in November 1953.[29][30] Telecommunication The concept of using a stored-program computer for switching of telecommunication circuits is called stored program control (SPC). It was instrumental to the development of the first electronic switching systems by American Telephone and Telegraph (AT&T) in the Bell System,[31] a development that started in earnest by c. 1954 with initial concept designs by Erna Schneider Hoover at Bell Labs. The first of such systems was installed on a trial basis in Morris, Illinois in 1960.[32] The storage medium for the program instructions was the flying-spot store, a photographic plate read by an optical scanner that had a speed of about one microsecond access time.[33] For temporary data, the system used a barrier-grid electrostatic storage tube. See also Stored program control References

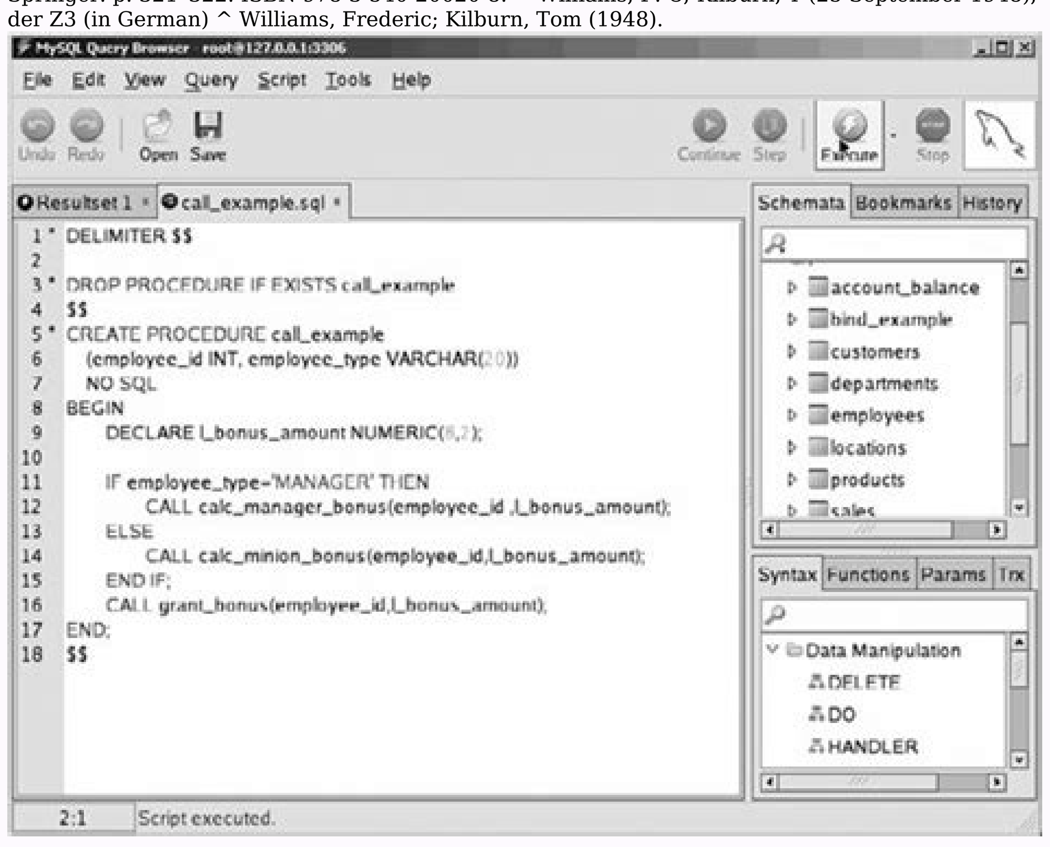
^ Allison, Joanne (1997), Stored-program Computers, archived from the original on 27 September 2011, retrieved 24 August 2011 ^ William F. Gilreath; Phillip A. Laplante (2003). Computer Architecture: A Minimalist Perspective. Springer. p. 24.
ISBN 978-1-4020-7416-5. ^ a b Edwin D. Reilly (2003). Milestones in computer science and information technology. Greenwood Publishing Group. p. 245. ISBN 978-1-57356-521-9. ^ Murdocca, Miles J.; Vincent P.



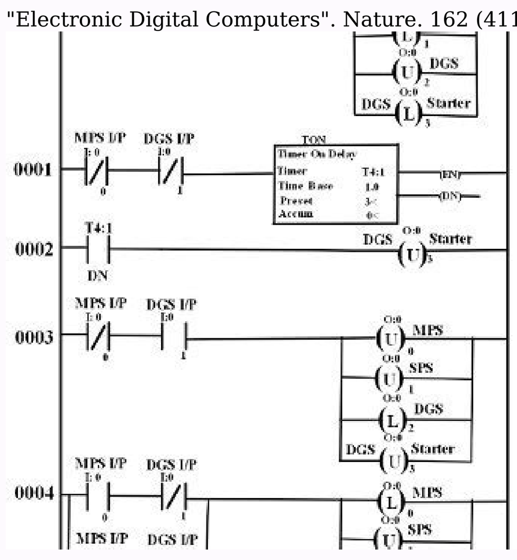
Heuring (2000). Principles of Computer Architecture. Prentice-Hall. p. 5. ISBN 0-201-43664-7. ^ Daniel Page (2009). A Practical Introduction to Computer Architecture. Springer. p. 148. ISBN 978-1-84882-255-9. ^ Mark Balch (2003). Complete digital design: a comprehensive guide to digital electronics and computer system architecture. McGraw-Hill Professional. p. 149. ISBN 978-0-07-140927-8. Retrieved 18 May 2011. ^ Daniel Page (2009). A Practical Introduction to Computer Architecture.



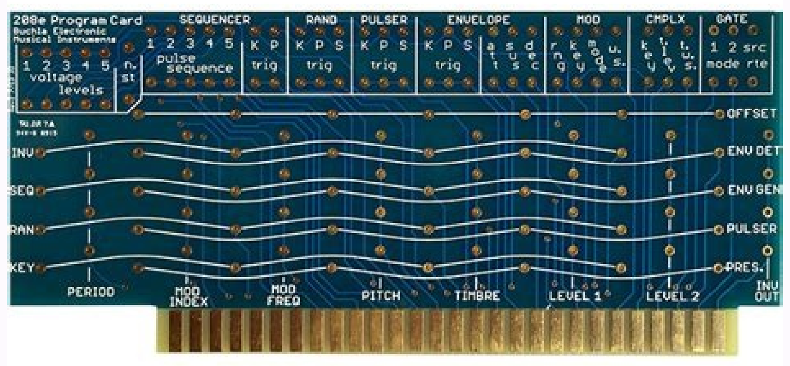
Springer. p. 153. ISBN 978-1-84882-255-9. ^ Ivor Grattan-Guinness (2003). Companion encyclopedia of the history and philosophy of the mathematical sciences. JHU Press. p. 705. ISBN 978-0-8018-7396-6. ^ Copeland, Jack (2000), A Brief History of Computing: ENIAC and EDVAC, retrieved 27 January 2010 ^ John L. Hennessy; David A. Patterson; David Goldberg (2003). Computer architecture: a quantitative approach. Morgan Kaufmann. p. 68. ISBN 978-1-55860-724-8. ^ B. Jack Copeland (2006). Colossus: the secrets of Bletchley Park's codebreaking computers. Oxford University Press. p. 104. ISBN 978-0-19-284055-4. ^ Christof Teuscher (2004). Alan Turing: life and legacy of a great thinker.
Springer. p. 321–322. ISBN 978-3-540-20020-8. ^ Williams, F. C; Kilburn, T (25 September 1948), "Electronic Digital Computers", Nature, 162 (4117): 487, Bibcode:1948Natur.162..487W, doi:10.1038/162487a0, S2CID 4110351, archived from the original on 6 April 2009 ^ Faber, Susanne (2000), Konrad Zuses Bemühungen um die Patentanmeldung der Z3 (in German) ^ Williams, Frederic; Kilburn, Tom (1948).



"Electronic Digital Computers". Nature. 162 (4117): 487. Bibcode:1948Natur.162..487W. doi:10.1038/162487a0. S2CID 4110351. Archived from the original on 6 April 2009. ^ Raúl Rojas; Ulf Hashagen (2002). The first computers: history and architectures. MIT Press. p. 379. ISBN 978-0-262-68137-7. ^ Daniel Page (2009).



A Practical Introduction to Computer Architecture. Springer. p. 158. ISBN 978-1-84882-255-9. ^ Mike Hally (2005). Electronic brains: stories from the dawn of the computer age. National Academies Press. p. 96. ISBN 978-0-309-09630-0. ^ Emerson W. Pugh (1995). Building IBM: shaping an industry and its technology. MIT Press. p. 136. ISBN 978-0-262-16147-3. ^ Olley, A. (2010). "Existence Precedes Essence - Meaning of the Stored-Program Concept" (PDF). History of Computing. Learning from the Past. IFIP WG 9.7 International Conference, HC 2010. IFIP Advances in Information and Communication Technology. Vol. 325. pp. 169–178. doi:10.1007/978-3-642-15199-6_17. ISBN 978-3-642-15198-9. ^ Graham, Loren R. (1993).
Science in Russia and the Soviet Union: A Short History.



Cambridge University Press. p. 256.
ISBN 9780521287890. ^ Emerson W. Pugh; Lyle R. Johnson; John H. Palmer (1991). IBM's 360 and Early 370 Systems. MIT Press. p. 15. ISBN 978-0-262-51720-1. ^ Thomas Haigh; Mark Priestley; Crispen Rope (2016). ENIAC in Action:Making and Remaking the Modern Computer. MIT Press. pp. 153, 157, 164, 174, 194. ISBN 978-0-262-03398-5. ^ Epic Technology for Great Justice – ENIAC ^ Campbell-Kelly, Martin (April 1982). "The Development of Computer Programming in Britain (1945 to 1955)". IEEE Annals of the History of Computing. 4 (2): 121–139. doi:10.1109/MAHC.1982.10016. S2CID 14861159. ^ Lavington, Simon, ed. (2012). Alan Turing and his Contemporaries: Building the World's First Computers. London: British Computer Society.
p. 61. ISBN 9781906124908. ^ Johnson, Roger (April 2008). "School of Computer Science & Information Systems: A Short History" (PDF). Birkbeck College. University of London. Retrieved 23 July 2017. ^ Hally, Mike (2005). Electronic Brains (First ed.). Granta. pp. 40–41. ISBN 978-1862076631. ^ Kilburn, T; Grimsdale, R L; Webb, D C (April 1956). "A transistor digital computer with a magnetic drum store". Proceedings of the IEE - Part B: Radio and Electronic Engineering. Cambridge University Press.
103 (35): 390–406. doi:10.1049/pi-b-1.1956.0079. ISSN 2054-0434. ^ Grimsdale, R L (Autumn 1995). "The Transition from Valves to Computers". Resurrection. Computer Conservation Society (13).
ISSN 0958-7403. ^ Carbaugh, D.H.; Marselos, N.L. (1983). "Switching System Software". In McDonald, J. C. (ed.). Fundamentals of Digital Switching Systems. Plenum Press.
ISBN 0-306-41224-1. ^ Joel, A. E. (October 1958). "An Experimental Electronic Switching System" (PDF). Bell Laboratories Record. 36 (10): 359–363. Retrieved 13 October 2022. ^ "Electronic Central Office". Long Lines. Vol. 40, no. 5. December 1960. p. 16. Retrieved from " Technology for telephone exchanges Stored program control (SPC) is a telecommunications technology for telephone exchanges. Its characteristic is that the switching system is controlled by a computer program stored in a memory in the switching system. SPC was the enabling technology of electronic switching systems (ESS) developed in the Bell System in the 1950s, and may be considered the third generation of switching technology. Stored program control was invented in 1954 by Bell Labs scientist Erna Schneider Hoover, who reasoned that computer software could control the connection of telephone calls.[1][2][3] History Proposed and developed in the 1950s, SPC was introduced in production electronic switching systems in the 1960s. The 101ESS private branch exchange (PBX) was a transitional switching system in the Bell System to provide expanded services to business customers that were otherwise still served by an electromechanical central office switch. The first central office switch with SPC was installed at Morris, Illinois, in a 1960 trial of electronic switching, followed by the first Western Electric 1ESS switch at Succasunna, NJ in 1965. Other examples of SPC-based third-generation switching systems include the British GPO TXE (various manufacturers), Metaconta 11 (ITT Europe), and the AKE, ARE. Pre-digital (1970s) versions of the AXE telephone exchange by Ericsson and Philips PRX were large-scale systems in the public switched telephone network (PSTN). SPC enables sophisticated calling features.

As such exchanges evolved, reliability and versatility increased. Second-generation exchanges such as Strowger, panel, rotary, and crossbar switches were constructed purely from electromechanical switching components with combinational logic control, and had no computer software control. The first generation were the manual switchboards operated by attendants and operators. Later crossbar systems also used computer control in the switching matrices, and may be considered SPC systems as well. Examples include the Ericsson ARE 11 (local) and ARE 13 (transit), as well as the North Electric NX-1E & D Switches, and the ITT Metaconta 11, once found throughout Western Europe and in many countries around the world. SPC technology using analog switching matrices was largely phased out in the 1980s and had disappeared from most modern networks by the late 1990s.

The addition of time-division multiplexing (TDM) decreased subsystem sizes and dramatically increased the capacity of the telephone network. By the 1980s, SPC technology dominated the telecommunications industry. Viable, fully digital switches emerged in the 1970s, with early systems, such as the French Alcatel E10 and Canadian Nortel DMS series going into production during that decade. Other widely adopted systems became available in the early 1980s.

These included Ericsson AXE 10, which became the world's most popular switching platform, the Western Electric 5ESS used through the US and in many other countries, the German designed Siemens ESWD, the ITT System 12 (later rebranded Alcatel S12) and NEC NEAX all of which were widely used around the world. The British developed System X (telephony), and other smaller systems also emerged in the early 1980s. Some digital switches, notably the 5ESS and very early versions of Ericsson AXE 10, continued to use analog concentrator stages, using SPC-like technologies, rather than direct connections to the digital line cards containing the CODEC. Early in the 21st century the industry began using a fifth generation of telephony switching, as time-division multiplexing (TDM) and specialist hardware-based digital circuit switching is replaced by softswitches and voice over IP VoIP technologies. The principal feature of stored program control is one or multiple digital processing units (stored-program computers) that execute a set of computer instructions (program) stored in the memory of the system by which telephone connections are established, maintained, and terminated in associated electronic circuitry. An immediate consequence of stored program control is automation of exchange functions and introduction of a variety of new telephony features to subscribers. A telephone exchange must run continuously without interruption at all times; it implements a fault-tolerant design. Early trials of electronics and computers in the control sub systems of an exchange were successful and resulted in the development of fully electronic systems, in which the switching network was also electronic. A trial system with stored program control was installed in Morris, Illinois in 1960.

It used a flying-spot store with a word size of 18 bits for semi-permanent program and parameter storage, and a barrier-grid memory for random access working memory.[4] The world’s first electronic switching system for production use, the No.1 ESS, was commissioned by AT&T at Succasunna, New Jersey, in May 1965. By 1974, AT&T had installed 475 No. 1ESS systems. In the 1980s SPC displaced electromechanical switching in the telecommunication industry, hence the term lost all but historical interest. Today, SPC is an integral concept in all automatic exchanges, due to the universal application of computers and microprocessor technology. The attempts to replace the electromechanical switching matrices by semiconductor cross-point switches were not immediately successful, particularly for large-scale exchange systems. As a result, many space-division switching systems used electromechanical switching networks with SPC, while private automatic branch exchanges (PABX) and smaller public exchanges used electronic switching devices. Electromechanical matrices were replaced in the early 21st century by fully electronic devices. Types Stored program control implementations may be organized into centralized and distributed approaches. Early electronic switching systems (ESS) developed in the 1960s and 1970s almost invariably used centralized control. Although many present day exchange design continue to use centralized SPC, with advent of low cost powerful microprocessors and VLSI chips such as programmable logic array (PLA) and programmable logic controllers (PLC), distributed SPC became widespread by the early 21st century. Centralized control In centralized control, all control equipment is replaced by a central processing unit. It must be able to process 10 to 100 calls per second, depending on the load to the system.[citation needed] Multiprocessor configurations are commonplace and may operate in various modes, such as in load-sharing configuration, in synchronous duplex-mode, or one processor may be in stand-by mode. Standby mode Standby mode of operation is the simplest of a dual-processor configuration. Normally one processor is in standby mode. The standby processor is brought online only when the active processor fails. An important requirement of this configuration is ability of standby processor to reconstitute the state of exchange system when it takes over the control; means to determine which of the subscriber lines or trunks are in use. In small exchanges, this may be possible by scanning the status signals as soon as the standby processor is brought into action. In such a case only the calls which are being established at the time of failure are disturbed. In large exchanges it is not possible to scan all the status signals within a significant time. Here the active processor copies the status of system periodically into secondary storage. When switchover occurs the recent status from the secondary memory is loaded. In this case only the calls which change status between last update and failure are affected. The shared secondary storage need not to be duplicated and simple unit level redundancy would suffice. 1ESS switch was a prominent example. Synchronous duplex mode In synchronous duplex mode of operation hardware coupling is provided between two processors which execute same set of instructions and compare the results continuously. If mismatch occurs then the faulty processor is identified and taken out of service within a few milliseconds. When system is operating normally, the two processors have same data in memories at all times and simultaneously receive information from exchange environment. One of the processor actually controls the exchange, but other is synchronized with the former but does not participate in the exchange control. If a fault is detected by the comparator the processors are decoupled and a check-out program is run independently to find faulty processor. This process runs without disturbing the call processing which is suspended temporarily. When one processor is taken out then the other processor operates independently. When the faulty processor is repaired and brought in service then memory contents of the active processor are copied into its memory and the two are synchronized and comparator is enabled. It is possible that a comparator fault occurs only due to transient failure which is not shown even when check out program is run. In such case three possibilities exists: Continue with both processors: This is based on the assumption that the fault is transient and may not appear again. Take out the active processor and continue with the other. Continue with active processor but remove other processor from service. When a processor is taken out, it is subjected to extensive testing to identify a marginal failure. Load-sharing mode In load-sharing operation, an incoming call is assigned randomly or in a predetermined order to one of the processors which then handles the call right through completion. Thus, both the processors are active simultaneously and share the load and the resources dynamically. Both the processors have access to the entire exchange environment which is sensed as well as controlled by these processors. Since the calls are handled independently by the processors, they have separate memories for storing temporary call data. Although programs and semi permanent data can be shared, they are kept in separate memories for redundancy purposes. There is an inter processor link through which the processors exchange information needed for mutual coordination and verifying the 'state of health’ of the other. If the exchange of information fails, one of the processors which detect the same takes over the entire load including the calls that are already set up by the failing processor. However, the calls that were being established by the failing processor are usually lost. Sharing of resources calls for an exclusion mechanism so that both the processors do not seek the same resource at the same time. The mechanism may be implemented in software or hardware or both. Figure shows a hardware exclusion device which, when set by one of the processors, prohibits access to a particular resource by the other processor until it is reset by the first processor. Distributed control Distributed SPC is both more available and more reliable than centralized SPC. The control function are shared by many processors within the exchange. It uses low cost microprocessors.

Exchange control may decomposed either horizontally or vertically for distributed processing.[5] In vertical decomposition the whole exchange is divided into several blocks and a processor is assigned to each block. This processor performs all tasks related to that specific block. Therefore, the total control system consists of several control units coupled together. For redundancy, processors may be duplicated in each block. In horizontal decomposition each processor performs only one or only some exchange functions. See also List of telephone switches Stored-program computer References ^ Alpha Doggs (2008-02-15). "Phone switching pioneers to be inducted in National Inventors Hall of Fame". Network World. Retrieved 2012-06-17. ^ "Erna Schneider Hoover". Maximumpc.com. 2012-06-17. Retrieved 2012-06-17. ^ "Erna Schneider Hoover". Global History Network of IEEE. 2012. Retrieved 2012-06-17. ^ Joel, A. E. (October 1958). "An Experimental Electronic Switching System".
Bell Laboratories Record. 36 (10): 359. ^ Thiagarajan., Viswanathan (1992). Telecommunication switching systems and networks. New Delhi: Prentice Hall of India Private Ltd. ISBN 0876927134. OCLC 29022605. Retrieved from "