Instruction format and addressing modes of sic

This article does not provide any sources. Please help improve this article by adding quotes to reliable sources. Non-sources of materials can be challenged and removed. Find sources: Simplified Computer Training - News Newspaper Book Scientist JSTOR (December 2009) (Learn how and when to remove this template message) Simplified computer (also abbreviated SIC) is a hypothetical computer system introduced into system software: Introduction to system programming, Leland Beck. Due to the fact that most modern microprocessors include subtle, complex functions for efficiency purposes, it can be difficult to learn system programming using a real system. A simplified computer solves this problem by abstracting these complex behaviors in favor of an architecture that is clear and accessible to those who want to learn system programming. The SIC Architecture SIC machine has the main address, storing most memory addresses in the six-core integrator format. Like most modern computing systems, the SIC architecture stores all the data in a binary version and uses it to represent negative values at the machine level. Memory storage in SIC consists of 8-bit bytes, and all memory addresses in SIC are byte addresses. Any three consecutive bytes form a 24-bit word, addressed to the location of the lowest number of bytes in the meaning of the word. Numerical values are stored as word values, and character values use an 8-bit ASCII system. The SIC does not support floating point equipment and has no more than 32,768 memory bytes. There is also a more sophisticated machine built on top of an SIC called a simplified learning computer with additional equipment (SIC/XE). The XE SIC extension adds a 48-bit type of floating point data, additional memory processing mode, and additional memory (1 megabyte instead of 32,768 bytes) to the original machine. The entire SIC up build code is compatible with SIC/XE. SIC machines have multiple registers, each 24 bits long and have both numerical and character representation: A (0): Used for basic arithmetic operations; known as the Battery Register. X (1): Stores and calculate addresses; known as the index register. L (2): Used to jump to certain memory addresses and store return addresses; known as the Link Register. PC (8): Contains the address of the following instructions to perform; known as the register program counter. SW (9): Contains various information such as flags carrying or overflowing; known as the status word register. In addition to the standard SIC registers, there are also four additional general-purpose registers specific to the SIC/XE: B (3): Used for the solution; known as the basic register. S (4): No special general-purpose register. T (5): No special use, general purpose register. F (6): Battery floating point point (This register is 48 bits instead of 24). These five/nine registers allow the SIC or SIC/XE to perform the simplest tasks in the custom-made Astra language. In System Software, this is used with a theoretical series of operations codes to help understand the collectors and connected downloaders needed to carry out the code of the asses language code. Address modes for SIC and SIC/XE Simplified Computer Instructions has three instruction formats, and additional hardware includes a fourth. The instructions provide a model for managing memory and data. Each format has a different view in memory: Format 1: Consists of 8 bits of dedicated memory to store instructions. Format 2: Consists of 16 bits of dedicated memory to store 8 bits of instructions and two 4-bit segments to store operands. Format 3: Consists of 6 bits for storing the instruction, 6 bits of flag values and 12 bits of offset. Format 4: Only SIC/XE machines are available, with the same elements as format 3, but instead of a 12-bit move, it stores a 20-bit address. Both format 3 and format 4 have six-bit flag values in them, consisting of the following bits of the flag: n: Indirect address flag i: Immediate address flag x: Indexed address address b: Basic address-relative flag p: Program counter-relative flag e: Format 4 instruction flag Address modes for SIC/XE Rule 1: e No 0 : format 3 e p No 0 (basic relative) b No 0, p No. p No 0 (direct address) format 4: b No 0, p No 0 (direct circulation) x 1 (index) i q 1, n No 0 (immediately) i q 0, n No 1 (indirect) i q 0, n No 0 (SIC) i q 1, n No 1 (SIC/XE for SIC compatible) Rule 2 : i 0, n No0 (SIC) b, p, e is part of the address. SIC Assembly Syntax SIC uses a special build language with its own operations codes that hold the hexagonal values needed to build and execute programs. An example of the program is below to get an idea of what a SIC might look like. There are three columns in the code below. The first column is a redirected symbol that will store its location in memory. The second column denotes either the SIC (opcode) instruction or the permanent value (BYTE or WORD). The third column takes the value of the character received when crossing the first column and uses it for the time of the operation specified in the second column. This process creates object code, and all object codes are entered into the object file, which will be controlled by the SIC machine. A COPY OF START 1000 FIRST STL RETADR CLOOP JSUB RDREC LDA LENGTH COMP ZERO JE'S ENDFIL JSUB WRREC J CLOOP ENDFIL LDA EOF STA BUFFER LDA THREE STA LENGTH JSUB WRREC LDL RETADR RSUB EOFTE BY C'EOF' THREE WORDS 3 ZERO WORD 0 RETADR RESW 1. A ROUTINE TO READ THE POST IN THE BUFFER. RDREC LDX ZERO LDA ZERO RLOOP ENTRY JE' RLOOP RD ENTRANCE COMP ZERO JE' EXIT STCH BUFFER, X X MAXLEN JLT RLOOP OUTPUT STX LENGTH RSUB INPUT BYTE X'F1' MAXLEN WORD 4096 . . A ROUTINE TO WRITE FROM THE BUFFER. WRREC LDX zero WLOOP TD OUTPUT JE'WLOOP LDCH BUFFER, X WD OUTPUT TIX LENGTH JLT WLOOP RSUB OUTPUT BYTE X'06' END FIRST If you were to collect this program, you will receive the object code pictured below. The beginning of each line consists of type and hex values for memory places. For example, the top line is the 'H' record, the first 6 six-digit digits represent its relative original location, and the last six six digits represent the size of the program. The lines are similar in all, with each 'T' entry consisting of 6 six-digit digits to indicate the original location of the line, 2 six-digit numbers indicating the size (in bytes) of the line, and the object codes that were created during the assembly process. HCOPY 0010000107A
T001001E141033482039001036281030154820613C100300102A0C1030390 001011E150C103648206108108108108101000004544000003000 T0020391E04103030000100001000100100205D30203FD8205D28103030302020555903C205E3333F
T0020571C1010364C0000F100010010000410302079302064509039DC20792C1036 T0020730738206434C00006 E001000 Example program below is a program illustrating the movement of data in SIC. LDA FIVE STA ALPHA LDCH CHAR'STCH C1 ALPHA RESW 1 FIVE WORD 5 CHAR' BYTE C'C1 RESB 1 SIC Emulation System Because SIC and SIC/XE machines are not real machines, the SIC emulator's task is often part of the SIC class coursework. The goal of SIC is to teach introductory system programmers or collegiate college students how to write and collect code below the higher level of languages such as C and C. With this being said, there are some sources of SIC-emulating programs over the Internet, however infrequently they may be. Assembler and simulator written by the author, Leland in Pascal is available on his educational home page at ftp permanent dead link SIC/XE Simulator And Assembler downloaded on the SIC Speaker Emulator, Assembler and some examples of programs written for SIC downloaded on the Tool Sics simulator, collector and link for the SIC/XE computer, available in See also computer MIX System Software Assembly Language Processor Register of Virtual Machines Links Beck, Leland (1996), System Software: Introduction to System Programming (3 ed.), Addison-Wesley, ISBN 0-201-42300-6 Info SIC and SIC/XE: //www-rohan.sdsu.edu/~stremler/2003_CS530/SicArchitecture.html List of SIC and SIC/XE Instructions: solomon/Course/SP.941/sic-instruction.html Brief Memory turning to SIC/XE Address: (permanent dead link) External links SICvm Virtual machine based on simplified training computer (SIC) derived from the SIC/XE, means simplified training equipment or additional equipment. This computer is a pre-version SIC. Both SIC and SIC/XE are closely related to each other, so they are compatible. SIC/XE: 1. Memory: Memory consists of 8 bits and 1 megabyte (220 bytes) in memory. The standard memory size of SIC is very small. This change in memory size leads to a change in the formats of the instructions as well as the address modes. 3 consecutive bytes form a word (24 bits) in the architecture of SIC/XE. All addresses are byte addresses and words are addressed by the location of their lowest-quality byte. 2. Registers: It contains 9 registers (5 SIC registers and 4 additional registers). Four Additional Registers: Mnmonics Use Register B Basic Register S General Work Register T General Work Register F Floating Battery Item 3. Data formats: Integrators are presented in binary numbers. The symbols are presented using ASCII codes. Floating dots are presented with 48-bits. 4. Instructions formats: The SIC/XE architecture has 4 types of available Bit (e) formats used to distinguish between formats 3 and formats 4, e'0 means Format 3 and e-1 means Format 4 Format 1 (1 byte): Format 2 (2 bytes): Format 3 (3 bytes): Format 4 (4 bytes): n'Indirect mode, i'Immediate addresses, x'Index addresses, b'Base addresses, program counter, e'Exponential address 5. Address modes: There is a use of the base register and the program counter to use the format 3. Target Address Indication Mode (TA) Basic Relative b'1, p.0 TAH (B) - Offset Program-counter b'0, p-1 TAH (PC) - moving a relative target address is an effective instruction address. 6. Instructions: In SIC/XE, all instructions are the same as those of the SIC architecture, but because of the floating point data format, it provides floating point arithmetic functions. To perform floating points of arithmetic operations, ADDF - Add floating points, SUBF - Subtract floating points, MULF - Multiply floating points, DIVF - Divide floating points SVC (Supervisor call) is also provided in the SIC/XE architecture for interrupting processing. 7. Entry and exit: The SIC/XE architecture includes I/O channels that allow I/O operations to perform while the processor performs other tasks. This will block the calculations and the vi, making it more efficient. Instructions such as SIO, TIO, HIO are used to run, test and stop I/O channels! Don't stop learning now. all-important CS theory concepts for SDE SDE With the CS Theory course at a student-friendly price and become a ready industry. Recommended messages: If you like GeeksforGeeks and would like to contribute, you can also write an article using contribute.geeksforgeeks.org or send your article by mail contribute@geeksforgeeks.org. See your article by appearing on the GeeksforGeeks Homepage and help other Geeks.Please improve this article if you find anything wrong by clicking on the Improve article below. Below. explain the instruction format and addressing modes of sic