Load store architecture examples

What is load store architecture.

These sÃ £ details of the MIPS R2000 architecture. The objective of this Ã © give the flavor of how all architectures were designed / specified from Ignatius' 80s different Ã Pentium. Load / Store Architecture ---------------------- The memory accesses to reduce a processor. Of course, we have added caches, but that brings the time-Only mÃ © dio access to memory accesses. HÃ¡ still times when the sampler processor the £'re doing nothing while waiting for the memory accesses to complete. Enta £ o, we define a new architecture. In this new architecture load / store, endereÃ§amento mode for each operating Ã © corrected. (So, do the £ hÃ¡ bytes to endereÃ§ar My Information mode.) And to Aritma type instructions © optical / lÃ³gico the endereÃ§amento mode for all operands serÃ¡ the record mode . Make sure that there are enough records, because everything ends up in records. For things to / from memory, and in / out records, we explÃcitas instructions that move data. Load instructions Read data from memory and copy it to a record. Store instructions write data from one record to the memory. The MIPS R2000 Ã © a load of architecture / shop. Records --------- There are 32 32-bit registers (for operand floating point in the £). (. The numbers sÃ £ 32-bit supplements the addresses sÃ £ 32-bit unsignados) in CÃ³digo, the syntax used Ã © $ x, where x © a value in the range 0 to 31. Example: $ 5 Ã © record 5. Some sÃ £ o special Goal: $ 0 account © mo always 0. This has the effect of reducing the instructions of how to Number in set of instructions. $ 1 Used by the assembler $ 2-3 Great location for the £ FunÃ§Ã Return values £ the $ 4-7 Great location for the £ ¢ meters to $ 26-27 Used exclusively by the OS $ 31 return ADDRESS Here placed §o procedures tamba sÃ © m £ 16 64-bit registers used for Ã floating point operands. In CÃ³digo, the syntax used Ã © $ fx where x Ã © par a value in the range 0 to 30. Example: $ F6 Ã © register 6. Some of the floating point registers Tamba © m tÃªm Specifics uses. Set of instructions to three addresses ---------------------------- The set of instructions Ã © considerably lower the sets of instructions from Intel. This Ã © tÃpico a RISC processor. Generally, there are a way to make a £ operaÃ§Ã the Required, do £ 2 or 3 as the Pentium. Instructions Aritma © optical and lÃ³gicas tÃªm 3 operands. Example: Add $ 8, $ 9, $ 10 the contents of the records of US $ 9 and US $ 10 sÃ £ o added, and the result Ã © put in $ 8 log To do the equivalent of this Pentium, you end up with something more like: mov eax, eax Add Var1, Var2; The instructions EAX set architecture (ISA) Ã © part of the processor that © Visible for programmer or compiler writer. The ISA serves as the boundary between software and hardware. briefly describe the sets of instructions found in many of the microprocessors used â â today. The ISA of a processor can be described using Catagories 5: Storage of operating the CPU, where the operands held Ala sÃ £ © m from memory? explÃcitos operands called Number how many operands the named sÃ £ instruction in a £ the tÃpica. Great location of the £ operating any operating instruction the ALU £ be located in memory? Or all operands must be kept internal to the CPU? OperaÃ§Ãµes which operaÃ§Ãµes sÃ £ provided in the ISA. Type and size of operands What Ã © the type and size of each operating and how it Ã © specified? Above all, the most distinctive factor Ã © the first. The three most common types of ISAS sÃ £ o: stack - the operands is £ implicitly on the stack. Accumulator - an operand Ã © implicitly the accumulator. general purpose register (GPR) - All operands sÃ £ explicitly mentioned, they sÃ £ the records or memory locations. Let's look at the mounting CÃ³digo C = A + B; In all three architectures: GPR battery stack push a load r1 one charge, a push Add BD R1, B Add Store C R1, C POP C - - Not all processors can be marked in one of the categories above. The I8086 has many instructions that use impliccit operands, although it has a general registration set. The I8051 is another example, has 4 GPRS banks, but Instructions should have a record of one of your operands. What are the advantages and disadvantages of each of these approaches? Stack Advantages: Simple Model of Expression Evaluation (Reverse Polishing). Short Instructions. Disadvantages: A stack can not be accessed randomly this makes it difficult to generate efficient code. The first pile is accessed to each operation and becomes a bottleneck. Advantages of the accumulator: short instructions. Disadvantages: The accumulator is only temporary storage so that the memorial traffic is higher for this approach. GPR advantages: makes the generation of easy codigo. Data can be stored for long periods in records. Disadvantages: All operands should be named leading to longer instructions. Previous CPUs were of the first 2 types, but in the last 15 years all CPUs were GPR processors. The two main reasons are that the records are more fast than the memory, more data that can be kept internal in the most fast CPU the program will be performed. The other reason is that the records are more convenient for a compiler to use. Reduced Instruction Computer (RISC) as we mentioned before the majority of modern CPUs are of the GPR type (general registration of proposed). Some examples of such CPUs are IBM 360, DEC VAX, Intel 80x86 and Motorola 68xxx. But while these CPUs were clearly better than previous cell cpus and accumulator, they were still missing in various areas: the instructions were of variable length of 1 byte at 6-8 bytes. This causes problems with the properties and the inclination of instructions. ALU (Logic Arithmetic Unit) may have operands that were memorial sites. Because the number of cycles leads to access the memory varies, so does all the instruction. This is not good for compiler writers, pipelining and multiple problems. Most Alu's instructions had only 2 operands where one of the operands is also destiny. This means that this operand is destroyed during operation or should be saved before somewhere. Thus, at the beginning of the 80's, the idea of RISC was introduced. The SPARC project was started in Berkeley and the MIPS project in Stanford. RISC means reduced instruction adjustment computer. ISA is composed of instructions that all have exactly the same size, usually 32 bits. So they can be proprietary and pipelined succession. All thread instructions have 3 operands that are just records. The only access to the memory is through explanatory instructions of load / store. So C = A + B will be assembled as: Charge R1, a load R2, B Add R3, R1, R2 Store C, R3 Although 4 instructions are required, we can reuse the values in the records. Why is this architecture called RISC? What is reduced about it? The answer is that to make all the instructions of the same size as the number of bits that are used â € â € âcode is reduced. Thus, fewer instructions are provided. The instructions that have been expelled are the less important operations of characters and BCD (decimal coded by binario). In fact, now that memoria access is restricted, there are not several types of movements or add instructions. Thus, the oldest architecture is called CISC (computer set of complete instructions). RISC architectures are also called load / store architectures. The number of records in RISC is usually 32 or more. The first RISC CPU MIPS 2000 has 32 GPRS in opposition to 16 in the 68xx and 8 architecture in the 80x86 architecture. The only disadvantage of RISC is the size of your code. Usually more instructions are needed and there is a waste in short instructions (pop, push). So why are you still being developed CCISC CCIS? Why is Intel spending time and money to manufacture Pentium II and Pentium III? The answer is simple, compatibility with verse. Compatible PC with IBM is the most common computer in the world. Intel A CPU that executed all applications that are in the hands of more than 100 million users. On the other hand, Motorola, which constructs the 68xx series that was used in the Macintosh made the transition and along with IBM and Apple, built the PPU (PPC), a RISC CPU, which is Top the new Power Macs. From now Intel and PC makers are £ o making more money, but with Microsoft playing on RISC Tamba field © m (Windows NT Ã © runs on Alpha Compaq) and the Java promise, the future of £ CISC during the course estÃ¡. A £ liÃ§Ã how important it can be learned here Ã © that superior technology Ã © indÃºstria a factor in computers, but as well © m Ã © marketing and price tag Tamba © m (on the £ else). ReferÃªncias for cell further reading Computing: The New Wave, Philip J. Koopman, JR, 1989 Philip Koopman, Jr. Philip Koopman Stack Computer Pages 5.5.1. £ standings the instruction by the £ Operands Memory-to-Memory (Vax, Architectures of sÃ © rie PDP) allow more than one operand of most instructions to access the memory. They usually allow ATA © three operand instruction by the £, all of which can be data records or addresses of memory. Addl3 A, B, SUM Register-Memory-Memory (X86, 68K) architectures allow only one operand of most instructions to access the memory. The other operand must be a CPU register. Most registration memory of CPUs on £ allows more than two operands for instruction £ o. In comparaÃ§Ã £ with a memory to memory, these processors cost less. Individual instructions sÃ £ o run faster due to fewer memory accesses, but it takes more instructions to accomplish the same task. MOVL EAX, EAX one ADDL, B MOVL SUM, EAX load-store architectures (MIPS, SPARC SPARC braÃ§o) on £ allow the most instructions access the memory. Only load and store instructions, which simply move data between registers and memory, can access the memory. All other instructions get their operands and store their results in records. Most instructions in lodagem CPUs Ã © performed very rÃ¡pida (often A single clock cycle), since they operate entirely within the CPU. However, lodagem architectures require the longest seqÃ¼Ãªncias of instructions to perform a particular task, since Ã © Necessary instructions to load separate the operands of memory, run operaÃ§Ãµes as £ adiÃ§Ã the, the subtraction £ etc. and store the results in memory. Note, however, that intermediÃ¡rios results in the £ tÃªm to be stored in memory at all, Enta £ o £ the situaÃ§Ã the sampler £ o Ã © tÃ £ o grisly as it may seem at first glance. Most stores architectures has a large Number of CPU registers for access to memory can be minimized. T0 $ LW, LW $ T1, T2 B ADD $, $ T0, T1 SW $ $ T2, accumulator-based base A.K.A. Architectures of an operand (8051, 68HC12) allow only one operating instruction for the £. The other operand where sÃ £ necessÃ¡rios the two operands, Ã © a special record called accumulator and estÃ¡ implÃcito. Most early processors was battery-based, and many embedded processors inexpensive today use this architecture. The Ld Add B Stum 5.5.2. £ standings by the memory structure in a von-Neumann architecture, A single address space of the memory © m account the instructions of the Chisel £ Execution programs as well as data . In such a system, there are just a memory address of 0, 1, 2 and so on. This architectural style Ã © named aft John Von Neumann, one matemÃ¡tico hÃºngaro American. Most end computers use this architecture. In a Harvard architecture, mÃ¡quinas CÃ³digo of the ES data and the £ stored in separate memory units connected to different busses. Because there are two (or more) of address spaces in memory, there are two AC © of memory cells with address 0 (CÃ³digo in one of the memory and a data memory), two with address 1 and so on. The Harvard architecture Ã © freqÃ¼entemente used for microcontrollers, where the program (called firmware) Ã © stored in an EEPROM in the £ volÃ¡til such as flash memory, so that when the power permaneÃ§a estÃ¡ off. The volÃ¡til RAM tradition £ o Ã © used only for data. Classification by complexity set of complex instruction computers (CISC) tend to have a large number of instruction (often hundreds) and memory organization for memory or memorial. Examples include the VAX, X86 and Series Motorola 68000 series. Many instructions require several clock cycles to complete, due in part to the ability to access the memorial. Set of reduced instructions (RISC) (RISC) Instructions (typically some diversions), and usually are the timing architectures. They tend to have a large number of records. Most instructions are performed on one or a few clock cycles. Cycles.