DOWNLOAD PDF SECOND EUROMICRO SYMPOSIUM ON MICROPROCESSING AND MICROPROGRAMMING, OCTOBER 12-14, 1976, VENICE

Chapter 1 : rodnay zaks : dÃ©finition de rodnay zaks et synonymes de rodnay zaks (anglais)

Second Euromicro symposium on microprocessing and microprogramming: October , , Venice Author: J Wilmink ; Mariagiovanna Sami ; European Association for Microprocessing and Microprogramming.

The theme of the conference was concentrated on the Architecture, Integration and Use of Systems with one or more microprocessors. Several sessions on architectures, both pro- cessor architecture as well as the architecture of multi- computer systems, were held as part of the conference program. Interface design was well re- presented with sessions in "Industrial Interfacing" and on "Bus Level Communication". One of the new sessions at the conference was on "Education", at which three relevant lec- tures were presented. Painke presented the main characteristics of the chip, both logical and physical and provided an example which highlights the tradeoffs between hardware and firmware im- plementation typical of the design. The second Keynote Address was presented by M. A Next Generation Microprocessor", was published in full in Vol. Presentations of manufacturers were given as part of the Industrial Program, which was organized by D. The Symposium was organized by P. The recent rapid advances of the LSI technology with the trend of increasing speeds and falling costs-per- chip have raised the interest in novel building blocks for use in memory systems. Associative memory is one example covered by the lecture in this session. Associative or Content-Addressable memories have fascinated computer designers for a long time because of their distributed processing capabilities. According to Vrielink, Information Retrieval, Vir- tual Memory Management and Computer Language Processing are some of the application areas where the need exists for simple processing of multiple data elements which are interrelated within a defined data structure. A Novel Associative Memory D. Tavangarian University of Dortmund, F. Each associative memory operation like writing, multi writing, reading out and parallel search, together with multi match resolving, is ex- ecutable in one cycle. Tavangarian explained that the resolution of multiple matches is achieved by novel masking counter which generates addresses for the access of all words which are marked by a match-bit. The developed associative memory also fulfills the functions of an ordinary random access memory. The realization of the introduced associative memory circuit as an integrated circuit, continued Tavangarian, is not restricted by pin- limitation problems because the number of pins does not surpass the pin number of normal RAMs. The combination of the integrated chips as a bus- oriented memory array is possible by means of an additional special chip select circuit and a chip in- dicating encoder. This feature allows the construc- tion of a large memory array as an associative storage unit. Industrial Interfacing Interfacing microcomputer based equipment so that it can perform its required task in an industrial L. Thompson Hawker Siddely Dynamics Eng. The real world of engineer- ing is hard and the environment is harsh. The pro- blems are aggravated by the behavior of people. Those associated with the use of the equipment so often do the unexpected and irrational. All this has to be achieved for the allowable price and within the allocated time scale. At that time when digital control depended upon the expensive cen- tralized computer, the problems of protection, communication and time sharing put the equip- ment at the mercy of the environment. Although the programmers and engineers made valient at- tempts, many installed systems proved unreliable and digital control got a bad name with production managers. The advances in technology in the last decade, particularly the availability of microprocessors of high capability, have completely changed the scenario for control and systems engineers. They can afford to partition their Systems and distribute the functions by using a multiplicity of processors. They can protect their circuitry in enclosures and include suitable filters in the connections so that sensitive devices are not subjected to the vigors of the environment. Software techniques are available so that systems can be tackled and operated in real time at a complexity level unthinkable only a few years ago. However, contended Thompson, the availability of capability has brought with it other dangers. The older people who

had learned by bitter experience what actually happens in an industrial environment are often not at home with the ap- proach and jargon of the new technologies while the bright new designers of hardware and software cannot communicate happily with those wo do not always follow their mental gyrations! So a gulf is growing, Thompson fears, but none of us can af- ford the time and the cost for the new generation to learn about environmental problems by experi- ment. The gulf has to be bridged. We need to en- sure that experience is built into the system while allowing the ingenuity of new workers full rein. The real life environment has not changed, The designer still has to have satisfactory answers to questions such as: Can I decide what are sensible levels for such phenomena? Can I accept the risk? How will my design react to irrational operator behavior? They were both prepared by engineers in the large Dutch electro- technical contracting company Van Rietschoten and Houwens, who have a wide experience in the application of new technology control systems in industrial complexes. He continued with the consideration of input signals in the industrial context and how their special problems can be solv- ed. Van Gennip treated the following subjects in detail: He brought out the essential need for discipline in design and the need to conform to stan- dards which have evolved from the experience of many people. Bouma touched on many practical matters ranging from how to arrive at a sensible performance specification to particular technical suggestions for combatting the environment. Bus Level Communication Chairwoman: M SamL Italy Intra-Digital-System Communication The availability of very large scale integrated circuits sug- gests the distribution of pro- cessing capability within an electronic system, therefore requiring efficient and simple protocols for intra-system communication. Brof- ferio Politecnico di Milano, Italy and G. Functional characteristics such as preemption, responsiveness and timing; physical performances as area, speed and power were defin- ed and discussed. These allow specification of the communication process and its structure to be systematically introduced. Trade-offs in performance are reflected in different solutions as the parallel and serial busses, which were analyzed in detail, sug- gesting suitable priority conflicts management, control signalling and timing. Brofferio and Scir6 noted that microcomputer busses can be described according to their proposed classification. As an application of the general concepts and rules in- troduced, an Asynchronous Circular Bus structure was defined and described in detail as part. Del Corso and F. Maddaleno Politecnico di Torino, Italy presented a technique to introduce into bus-based systems new hard- ware primitives for information exchange. The starting point, explained F. It consists in a specific encoding of protocol signals into bus lines. An implementation of the extended protocol was described in detail. Maddaleno concluded the presentation by describing the design and fulfill- ment of a module that performs memory reloca- tion and protection by using the protocol extension technique. Today microprogramming establishes an important layer in the hierarchical structure of a computer system [3, 6, 9, The second pun has in the meantime lost some of its validity. It is still true that a large percentage of microcode firmware emulates com- puter architectures but the availability of writeable control stores has brought about a completely new usage. It has become easier and more common to migrate functions from software into firmware. For both disciplines the same objectives are of importance: For Soft- ware Engineering the above list means decreasing importance, whilst for microprogramming it means increasing importance. This allows Software Engineers to proceed to ever higher levels of abstraction e. Meta IV, [5] , while firmware engineering always has to consider the hardware characteristics early in the design process. They presented results with respect to vertical migration of func- tion in a real-time environment. After describing the goals of the migration process, H. Kaestner gave a short overview of the underlying real-time system - written in the language Modula. This was followed by a discussion about the selection mechanism that has been used to determine the candidates for migration. The main problems in connecting microprograms to a high-level language program was then investigated, i. Finally, Kaestner discussed the speed improvements that have been achieved by the migration process. For different classes of functions the local im- provements as well as the effect on the overall system performance was shown. Microprogram

DOWNLOAD PDF SECOND EUROMICRO SYMPOSIUM ON MICROPROCESSING AND MICROPROGRAMMING, OCTOBER 12-14, 1976, VENICE

Production Environments The second lecture of this session, presented by P. Norrie, produced a strong argument for developing complete microprogramming pro- duction environments. Some basic tools and methodologies are still inadequate or missing. The authors discussed some directions in which such a production environment should be developed. Principles of Firmware Engineering in Microprogram Control. Elements of Microprogram- ruing. Critical Remarks and a Proposed Strategy. Firmware, Microprogramm- ing and Restructurable Hardware, pp. The Vienna Develop- ment Method: Springer Lecture Notes in Computer Science, vol. Microprocessing and Microprogramming, vol. Firmware, Microprogramming and Restructurable Hardware, pp. Band 31, Springer Berlin Microprogramming - Principles and Prac- tices. Prentice Hall, Englewood CI. Models and Structures for Microprogramm- ing. Computer In- augural Conf. The following short notes, a few of which are published in full in this issue, were presented in the following two topical sessions: Development Environments and Applications Chairman: A Proposed Method L. Hannemyr, The Netherlands Applications Chairman: Azaria, Israel Education Chairman: Wilson, Polytechnic of Central London, U.

DOWNLOAD PDF SECOND EUROMICRO SYMPOSIUM ON MICROPROCESSING AND MICROPROGRAMMING, OCTOBER 12-14, 1976, VENICE

Chapter 2 : CiNii Books Author - EUROMICRO Symposium on Microprocessing and Microprogramming

Second Euromicro Symposium on Microprocessing and Microprogramming, October , , Venice: [papers].

This invention relates to a signal processing system specifically to a micro-vector multi-processor. Description of the Prior Art In the past, to efficiently perform vector arithmetic such as fast fourier transform, correlation and convolution, digital filtering transversal and recursive , quadrature translation, digital beam forming and integration and smoothing, large scale pipeline computers such as the Texas Instruments Incorporated Advanced Scientific Computer ASC were commonly employed. The AATP is a programmable peripheral processing unit designed and programmed to efficiently implement signal processing algorithms. However, such a system does not provide autonomous operation of a vector processor which shares memory with a minicomputer. The emphasis on this system is its parallelism and its ability to perform many operations dependent on each other in parallel through the use of many microprocessors. These microprocessors are data-driven which implies that the presence or absence of data determines whether a micro- processor will perform an instruction or not. This parallelism requires programming that must be peculiar to the structure. If the programming is done in the more conventional sequential manner the parallel structure provides no real advantage. It can be said that the system known from the before mentioned document performs a datade-pendent processing. In the document "Computer Design, Vol. Such an operation does not involve signal processing per se and certainly such an arithmetic unit does not anticipate vector hardware for signal processing. A main computer carefully monitors and controls the floatihg circuit. This invention provides fast and economical vector arithmetic and signal processing in general. Summary of the Invention In essence, a signal processing system as defined in claim 1 is introduced into a general purpose computer system such like a peripheral device, to autonomously perform signal processing algorithms on command from the general purpose computer. The micro-vector processor is an integrated circuit implementation of a digital signal processor. In this preferred embodiment, it is based, in part, upon the high density, low power semiconductor technology of integrated, injection- current logic WL. Schottky output devices replace resistors to further lower the power requirements. An input controller receives input signals and provides them, in digital form, for storage in a data memory unit. A programmable supervisor unit, whose function is dictated by a micro-processor has access to the data memory unit. The supervisor also controls the sequencing of the vector unit and controls the input and output controllers. The vector unit functions under control by a microcontroller whose programming is by way of a ROM. Algorithms for performing vector computations are stored in the ROM associated with the micro-controller. The VCU has the micro- controller and one half of the ROM and additionally has an arithmetic logic unit for updating addresses in the data memory. The CALU, in addition to having the other half of the ROM, has a fast multiplier and a pair of arithmetic logic units for performing the necessary arithmetic to implement the vector arithmetic algorithms contained in the ROM. As the various computations are completed, the results are sent to the data memory unit. When the overall vector computation is completed, the VCU notifies the supervisor of this fact and a new computation is initiated after the supervisor causes the data memory unit to read out the calculated data to the output controller. Supplement to the Description This system is contrasted with the systems involving interconnection between a large number of microcomputers or minicomputers wherein each computer is assigned a portion of a task and one computer is assigned the management responsibilities. When given real time signal processing problems the vector unit or units of this invention may provide results within several orders of magnitude faster than the system involving a plurality of microcomputers or minicomputers making a timely real time solution possible. In a system involving a plurality of microcomputers or minicomputers a response to the problem may well require a solution long before that system is able to provide such a solution. For applications involving for example

fast Fourier transforms only the system of this invention is worthwhile when a solution is needed immediately. The input controller receives incoming data which, in this preferred embodiment, is points of digital data converted from an analog input. The input controller has the capability of addressing the data memory so that the incoming data may be properly stored. The output controller, in this preferred embodiment, is identical to the input controller, having the ability to address the memory to read data from the memory and to output it. The data memory itself has a static, random access memory RAM which is controlled by a memory controller. It also has a word swap multiplexer which enables the interchange of the higher order bits of a double length word for the lower order bits in either the reading or writing mode. The memory control performs an incrementing function for the RAM for purposes of supplying a double length word and also determines which of the units, the supervisor, the vector unit, the input controller or the output controller, has priority in the event of simultaneous requests for use of the data memory. As mentioned earlier, the in part implementation of this preferred embodiment with respect to the microcontroller, register files, arithmetic logic units, and multiplier is in 12L. The 12L technology of this preferred embodiment is an improved technology, utilizing Schottky devices as the communication link outside of the 12L logic arrays. The Schottky circuitry further lowers the power consummed by the 12L circuitry. However, in this preferred embodiment, to even further reduce the power consummed, a power up-power down circuit is employed to cause units and components to be disconnected from the power source when not in operation. The object of this invention is to provide a micro-vector multiple processor having a vector unit capable of performing complex vector manipulations upon initiation by a general purpose microprocessor which does not have to intervene once the vector unit has been started and which may proceed on its own processing chores. Another object of the invention is to provide a microvector multiple processor having a single general purpose microprocessor controlling a plurality of vector units, each capable of performing complex vector computations independently of the microprocessor and each other, once having been invoked by the supervisor. These and other objects will be made evident in the detailed description that follows. FIGURE 2 is a block diagram of the micro- vector processor of this invention wherein there are four vector units and one supervisor unit. FIGURE 8 is a schematic diagram illustrating interrupt control generated from the address register of the microprocessor. FIGURE 10 is a schematic diagram of the address transceiver between the lower order address lines from the supervisor to the data memory. FIGURE 11 is a schematic diagram of the address transceiver for the higher order address lines from the supervisor to the data memory. FIGURE 13 is a schematic diagram showing the address transceiver for the bias register to connect to the data memory. FIGURE 14 is a schematic diagram of the data transceiver read data of the supervisor to the data memory. FIGURE 16 is a schematic diagram of the prioritizing circuitry and other control circuitry of the memory control. FIGURE 17 is a schematic diagram of the circuitry of the memory control for receiving and incrementing the input address to the data memory. FIGURE 24 is a schematic diagram of the primary and secondary data memory address and control interface. FIGURE 1 illustrates a data memory 11 connected by bus 15 to supervisor 12, vector unit 13, input controller 14 and output controller Vector unit 17 is connected via bus 25 to data memory section 21 and to data memory section 22 via bus Data memory section 22 is connected via bus 27 to vector unit 18 which is in turn connected to data memory section 23 through bus Bus 29 connects vector unit 19 to data memory section 23 and bus 30 connects vector unit 19 to data memory Data memory 24 is connected via bus 31 to vector unit FIGURE 2 illustrates vector units connected to a sectionalized data memory 11 or to separate data memories equivalent to sections for separate execution of different tasks in a series of algorithms to be performed. Memory control 33 also has address and control lines input FIGURE 4 illustrates supervisor 12 having address transceiver 41 connected to program memory 46 and bias register 43 via bus Data transceiver 42 is shown connected via bus 48 to program memory 46 and to microprocessor Microprocessor 44 is connected to interrupt control 45 which has a bidirectional bus Another output is shown going to address bus 37 of Fig.

Turning now to Figure 16, part of the memory control 33 is shown in detail. Clock input Phase 1 of 4 is shown applied to the input of inverter The system clock not shown is physically located on the supervisor unit and, in this preferred embodiment, is originated by a 24 MHz crystal from the Vectron Company, whose output is appropriately divided to provide a 4 MHz clock for the microprocessor of the supervisor unit and a 2 MHz clock for use in other components of the system. The 2 MHz frequency is decoded from a ring counter to form a 3-phase, 4-phase, and 5- phase clock. Having these multi-phase clocks available provides a facility of an available pulse when needed without having to wait a period of time until the next pulse is available. The clock system, as described, is well known in the prior art and need not be described in more detail here. They are also applied to AND gates , and , respectively. AND gates do not perform a logic function but are in the circuit to provide a delay equal to that of inverters NAND gate has an input from each of AND gates , and and also has an input from the output of inverter The clock output from inverter is inverted again through inverter as output serves as an input to each of NAND gates , , , and Exclusive OR gate has an input from jumper circuit and another input A3 from the address input to memory control Inputs A2 and A3 are provided to select other memory boards. That is, in this preferred embodiment, the static memory selected has a capacity of 32, bits, and other memory boards may be added. When added, they are selected by signals A2 and A3. Jumper circuit , when connected as shown to VCC, causes a high output from exclusive OR circuit when signal A2 is low. Conversely, when jumper circuit is connected to ground, the output of exclusive OR circuit is high when signal A2 is high. In the same manner, when jumper circuit is connected to VCC, the output of exclusive OR circuit is high when signal A3 is low. When jumper circuit is connected to ground, the output of exclusive OR circuit is high when input signal A3 is high. The outputs of exclusive OR circuit and are connected as inputs to NAND circuit whose output is inverted through inverter and applied as an input to AND circuit The output of inverter has phase 1 of 4 clock and is applied to the clock input of latch The output of inverter is inverted again through inverter and applied to the clock input of flip-flop The D input of flip-flop is supplied by the output of inverter Figure 17 continues with the description of the memory control Also input is Phase 3 of 4 clock to inverter whose output is the enable input to latch Address bit A6, on the Q3 output of latch is applied to the A1 input of adder Address bitA5, from output Q2 of latch is applied to input 82 of adder and also to the input of NAND circuit They are also applied as inputs to NAND circuit Adder is identical to adder Address bits A12, A13 and A14, present outputs on terminals Q1, Q2 and Q3, respectively, of latch and are provided as inputs A3, A2 and A1 of adder , which is identical to adder The carry generated by the addition of B1 to A1-A4 in adder caused by increment address bitA19 is generated and sent out on terminal C4to the carry input terminal CO of adder In like fashion, terminal C4 of is connected to terminal CO of adder to provide a carry, if any, to that adder. Similarly, terminal C4 of adder is connected to terminal CO of adder to provide a carry, if any, to adder The resultant sum from these adders is the original input to the adders incremented by one and designated summation AO-summation in reverse order. That is, summation 14 is present at summation 4 output of adder , while AO is present at summation 1 of adder Figure 18 illustrates the word SWAP multiplexer 35 with respect to the circuitry required to swap words being written into the memory array Output terminal 1Y of multiplexer presents either data bit B00 or data bit B18 depending upon the input to the S terminal which is the SWAP B input described earlier.

DOWNLOAD PDF SECOND EUROMICRO SYMPOSIUM ON MICROPROCESSING AND MICROPROGRAMMING, OCTOBER 12-14, 1976, VENICE

Chapter 3 : Aksenti Grnarov | South East European University

Rodnay Zaks (born 10 February , Paris) is a French-born American author of many books on computer programming, including the seminal Programming the Z80 and Programming the

A first memory from which data is normally read and a second memory into which data is normally written are assigned to the same address space. In a first mode of operation the first memory can be selected by a read a signal and the second memory by a write signal. In a second mode of operation the first and second memories are selected by write and read signals respectively. Arrangements are described where the first memory is ROM and the second memory RAM and in the second mode of operation the RAM is accessed for both reading and writing. The arrangements of the memories in the addressable memory space is alterable. Description The present invention relates to memory addressing systems. Typically, computer processors have a certain maximum number of bits with which to specify an address. This set of addresses defines an address space in which peripherals to the processor must be placed. Figure 1 shows a typical example of a memory map obtained by a typical allocation of processor address space. In total, the addressable memory is 64K bytes, the maximum number of bytes addressable with 16 bits of information. Recently, the length and complexity of software has markedly increased, so an address space of nearly 64K bytes may be required simply for operation of a program. The number of pixels included in a CRT display has also increased as the requirement for higher resolution of displays has increased. To store this increased number of pixels, an addressable memory space of nearly 16K bytes is needed for CRT refresh memory This is likely to be the minimum memory capacity that will be necessary for future graphic displays. However, the addressable space accessable by processor is physically limited by the number of address bits that can be handled. A conventional processor can handle 16 address bits which can individually address no more than 64K bytes of memory. This results in severe restrictions in system design. The present invention seeks to provide a memory addressing system which has a substantially enlarged addressable memory space. According to the present invention there is provided, a memory addressing system comprising: A number of preferred embodiments of the invention will now be described by way of example and with reference to the accompanying drawings, in which: Figure 1 shows a prior art memory map; Figure 2 is a block diagram of a memory addressing system; Figure 3 shows a memory map according to a first embodiment of the invention; Figure 4 is a circuit diagram of a memory selection circuit for the embodiment of Figure 3; Figure 5 shows a memory map according to another embodiment of the invention; Figure 6 is a circuit diagram of a memory selection circuit for the embodiment of Figure 5; Figure 7 shows a memory map according to yet another embodiment of the present invention; Figure 8 is a circuit diagram of a memory selection circuit for the embodiment of Figure 7; Figure 9 shows a memory map according to a further embodiment of the present invention; Figure 10 is a circuit diagram of a memory selection circuit for the embodiment of Figure 9; Figure 11 shows a memory map according to a still further embodiment of the present invention; and Figure 12 is a circuit diagram of a memory selection circuit for the embodiment of Figure Control line propagates a control signal generated by the CPU to memory control unit After memory control unit has received this signal through control line , it supplies memory control signals to ROM and RAM through control lines The details of the internal construction of memory control unit will be explained hereinafter. Figure 3 is a memory map showing a logical arrangement of memory in an embodiment of this invention. The addressable memory space of 64K bytes is divided into four areas of 16K bytes. The memories which are included in this addressable memory space differ depending on which of two possible modes mode "0" or mode "1" is preset. Changeover between these modes is accomplished by a mode selection circuit, which will be described later Figure 4. A technique, independent of CPU e. Consequently, normally, the address space is in mode "0" and the program stored in

ROM a and ROM b is performed through a series of read operations. In this way, even though the memory space is 64K bytes, the whole space can be used as program memory, and the 16K bytes of the CRT memory can also be used. Thus a total memory capacity of 80K bytes is available, so the available memory space has been effectively expanded. To accomplish reading, the address space is temporarily put into mode "1". Consequently, software may then be provided such that CPU carries out a read operation in a conventional manner. However, mode "1" operation does not normally require a long time, so that the mode changeover need not be performed long before RAM c is to be read. Thus software limitations can be easily overcome. An embodiment of a mode selection circuit for carrying out the mode changeover of the address space will now be explained with reference to Figure 4. In Figure 4, the uppermost address bits, A14 and A15 of the bit address, are supplied to decoder The enable terminal EN of decoder is supplied with a status signal that indicates whether the program step calls for a memory input or a memory output to be performed. It is well known that this is a signal that is generated at the beginning of every machine cycle, so that an explanation of it will be omitted. Decoder detects which memory or is to be accessed and outputs this information as selection signals SEL O - SEL 3 at the respective output terminals 0, 1, 2 or 3. The signal SEL 0 that is outputted from the output terminal 0 of decoder is supplied to the enable EN of selector A data signal DT1, which is indicative of the desired operating mode is applied to the D input terminal of mode selection flip-flop by an output command of CPU The Q output of mode selection flip-flop is supplied to the selector terminal SL of a gate selector The input terminal 1 of gate which is part of selector is, grounded, and the signal MRD that indicates that memory reading is in progress is supplied from the CPU to the input terminal 0. The input terminal 1 of gate of the selector is supplied with the output of OR gate , which receives as inputs the signal MRD and a signal MWR. The signal MWR indicates that a memory write operation is in progress. The selector enables either gate or in accordance with the content of mode selection flip-flop In this embodiment of the invention, the output of gate selects ROM a, and the output of gate is provided to RAM c. During normal operation, i. When the address bits A14 and A15 are both low, the SEL 0 output of decoder is enabled, thus enabling the selector As a result, the program is read out from ROM a. When RAM c is to be refreshed, a write cycle is required. As a result, the output of gate of the selector is enabled so that RAM c is selected. A writing operation is therefore performed on RAM c. In mode 1, the RAM c may be read so as to check its contents. Thus, in mode 0, when gate of selector is selected, since its input is given by the read-indicating signal MRD only, ROM a can only be read. This may be accomplished as in known systems, and an explanation thereof is therefore omitted. It will be appreciated that by means of the mode selection circuit if Figure 4, the entire 64K byte memory space can be utilized as program memory, and in addition, 16K bytes of RAM c can be accessed. This makes it possible for a total memory capacity of 80K bytes i. The above explanation was based on the address space of the memory map shown in figure 3, but the organization of the addressable memory space is not limited to this form. First a brief explanation will be given of the structure and operation of the mode selection circuit shown in Figure 6 using the truth table of Table 2 and the memory nap shown in Figure 5. The other mode, mode 1, is obtained when mode selection flip-flop is set. Instead, RAM c is placed in this area of the address space. Also in mode 1, the ROM a is accessable as normal. Thus, programs may usually be run normally even when, as in mode 0, the addresses overlap. In this way the memory space can be expanded. However, since the CPU may occasionally wish to read from RAM c, a way must be provided to read, for exampe, "what letter is being displayed" or "what kind of dot pattern is being displayed". Furthermore, the RAM c may be used as a message buffer. To allow for such cases, in mode 1 RAM c can be, placed in the address space occupied by RAM b in mode 0, so that RAM c can be read from and written in a normal fashion. The mode selection circuit of Figure 5 is an example of a circuit which can perform this form of control. The circuit of Figure 5 differs from the embodiment of Figure 4 in that the selector has three gate stages, , and which have different transfer functions as illustrated in Table 2. Figure 7

differs from Figure 3 by the substitution of memories and in the address space occupied by ROM b in Figure 3. The memory may be thought of as a program storage memory and the memory may be thought of as a display memory. The memory may be RAM, so that a program stored in memory may be changed. However, the memory normally contains a program, and the CPU normally reads the program and runs it, so in almost all cases memory need only be read. However, if the program content is to be changed, data must be written into memory The memory normally only has display data written into it by CPU , but sometimes it is necessary to check the letters or patterns that are being displayed, and in such cases a read operation is also necessary. Consequently, if memories and are simply permanently assigned to the same address space with fixed "READ" and "WRITE" operation respectively, the less common, although required functions of these memories could not be performed. Thus, the CPU must be able to read and write to both of the memories. An embodiment- which satisfies the above requirements is the memory map shown in Figure 7 and the mode selection circuit shown in Figure 8. This embodiment also has modes 0 and 1. In mode 0, memories and have their respective ordinary functions and occupy the same address space as read-only and write-only memories. In mode 1, a control is applied to change memory over to enable write operations and memory over to enable read operations. Thus, in mode 0, memories and operate so as to effectively increase the memory space, as if they were , separate memories. Yet memories and overlap on the same address space, with data being read from memory and data being written into memory However, in special cases when mode 1 is entered , these read and write functions are interchanged so that data can be written into memory and data can be read from memory The operation of the mode selection circuit illustrated in Figure 8 for controlling the memory configuration illustrated in Figure 7 will now be described with reference to Table 3. In normal operation, i. Consequently, the output of gate of the selector is "HIGH" and the output of gate is "LOW", so that the memory is selected and the program is read out. Consequently, the output of the gate of the selector is "LOW" and the output of the gate is "HIGH", so that the memory is selected and the write operation can be performed thereon. In mode 1, a program is loaded into memory or data stored in memory is read. Consequently, the output of gate of the selector is "HIGH" and the output of gate is "LOW" so that the memory is selected and the program is loaded into the memory The memory is therefore selected and the data stored in memory is read. A mode selection circuit for controlling the memory map of Figure 9 is shown in Figure 10 and the circuit for controlling the memory map of Figure 11 is shown in Figure The circuit shown in Figure 10 operates in accordance with the truth table shown in Table 4 below. Table 4 may be summarized as follows. The circuit shown in Figure 12 operates according to the truth table shown in Table 5, given below. Table 5 may be summarized as follows:

DOWNLOAD PDF SECOND EUROMICRO SYMPOSIUM ON MICROPROCESSING AND MICROPROGRAMMING, OCTOBER 12-14, 1976, VENICE

Chapter 4 : Activities | Miroslaw Malek

Second Euromicro Symposium on Microprocessing and Microprogramming, October , , Venice: [papers] edited by Jan Wilmink, Mariagiovanna Sami, Rodnay Zaks ; organized by Euromicro ; sponsored by Euromicro.

Having thus described our invention, what we claim as new, and desire to secure by Letters Patent is: A programming logic array PLA having an AND array and an OR array with input lines to the AND array, output lines from the OR array, and product term lines interconnecting the two arrays, the PLA having storage flip-flops at the cross points of the arrays, wherein the improvement comprises, means connecting the flip-flops along individual product term lines of the AND array and along individual output lines of the OR array to form shift registers, means for sequentially loading data for a test sequence into the shift registers of the PLA, whereby the array can be tested to locate faults, and dummy lines in the array connectable to replace a line in which a fault is detected. A PLA in accordance with claim 1, wherein the means for loading test sequence data comprises, a test storage and a load storage and a step counter for addressing the test storage and load storage, means responsive to the occurrence of an error in the PLA during a test to reset the counter to the beginning address for a test sequence. A PLA in accordance with claim 3, wherein the comparison evaluator comprises a number of indicator lines corresponding to the number of output lines, whereby, in the case of an error in the OR array, one and only of said lines indicates which output line is erroneous, and means for switching out the erroneous output line and switching in an OR dummy line is switched on instead. A PLA in accordance with claim 2, wherein the means for loading test sequence data comprises, p-control logic means and m-control logic means connected to receive control signals from the test storage, and means connecting the p-control logic means for column addressing of the AND array and connecting the m-control logic for addressing column and row addressing of the OR array. A PLA in accordance with claim 3, including a p-counter, a p-register connected to be loaded from the p-counter, and a p-decoder for addressing the AND array from the p-register, an m-counter, an m-register connected to be loaded from the m-counter, and an m-counter for addressing the OR array from the m-register, and means responsive to a test error for loading the p-register and the m-register from the p-counter and the m-counter. A PLA in accordance with claim 6, including a coder for converting the signal received on one of the indicator lines of the comparison evaluator into a coded representation of the respective line, and means connecting the output of the coder to a further input of the m-register. A PLA in accordance with claim 6, characterized in means connecting the p-counter to address the load storage for the AND array, a first shift register connected to receive the output, whose output, via the p-control logic, can be connected to the input of the shift register of the AND array indicated by the p-decoder, and in that the output of the m-counter addresses a load storage for the OR array, whose output is connected to a second shift register, the output signal of which, via the m-control logic can be connected to the input of the shift register of the OR array indicated the the m-decoder. Array in accordance with claim 8, characterized in that the output of the second shift register 92 is connected to the input of a first AND gate , FIG. Array in accordance with claim 9, characterized in that during the last load step of a shift register of the OR array, a first force signal BIT ON, table IV is generated, switching the output of the flip-flop circuit 96 , via the first OR gate , to the flip-flop circuit of the zeroth column of the OR array 7 in such a manner that the flip-flop circuits LO 00 to LO 30 of the OR array belonging to the zeroth column FIG. Array in accordance with claim 7, characterized in that for loading the dummy line OEL of the array 7 , the contents of the m-register 84 are transferred to the m-counter 86 , and that by a second force signal F OEL , a fourth AND gate is activated, switching the output of the second shift register 92 , via the fourth AND gate, to the input 30 of the shift register SRO, LO 00 to LO 04 belonging to the dummy line. Field of the Invention The invention concerns a programmable logic array PLA and, in particular, the testing and loading of a PLA made up of an

AND array and an OR array and in which the cross points in the arrays are controlled by storage flip-flop circuits. Signals corresponding to a number of binary variables are applied to the input of such a PLA, and, depending upon the programmed activation of the cross points in the two arrays, output signals are received on particular output lines function lines. A preferred application of such a PLA may be its use as a function control in a data processing system. For this purpose, the binary operation signals of an instruction are applied to the input of the PLA, and the data flow is controlled by means of the output signals on particular function lines. Large-scale integration LSI permits relatively large PLAs together with the storage flip-flop circuits to be realized on one chip. As a result, the output lines of the AND array, which simultaneously form the input lines to the OR array and which are referred to as "product lines", are no longer accessible from the outside. This makes the testing of a PLA far more difficult. Such testing may be effected during or after completion of manufacture or during the operation of the PLA in a data processing system. In such a test, the object is to determine whether a line or a transistor at the cross point of a matrix has become stuck at a single logic level. A line may be, for example, short-circuited to ground or to the supply voltage. The line may be interrupted, and in most cases an interrupted line assumes one of these potentials. If the nodes of the arrays, i. Apart from a few exceptions, it is not possible to test a PLA during its operations in a data processing system. So far, testing has been effected before installation of the PLA in the data processing system by applying special test signals and comparing the output signals with known desired values or testing after installation has been made by interrupting the normal operation and then using a diagnostic routine which also applies test patterns to the input of the PLA. The Prior Art With regard to prior art, attention is drawn, for example, to U. Hong and Daniel L. These known arrangements permit testing the PLA only when normal operation has been interrupted. Apart from this, these known methods are not suitable for a PLA with storage flip-flop circuits in the cross points, as they do not utilize the possibility of dynamic personalization characteristic of such a PLA. The complete testing of the PLA rather requires the application of a very great number of test signal combinations and thus a long test period. Muehldorf teaches a logic arrangement consisting of two series-connected arrays. The cross points of one of the arrays may be personalized by loadable flip-flop circuits, while the array matrix is permanently personalized. For testing, all flip-flop circuits of an array may be connected in the form of a single shift register, through which a test pattern is shifted. The base of each of the transistors is connected to the input line, and the emitter of each of the transistors is connected to the output line referred to as word line. For each word line there is a further transistor which becomes conductive when a cross point transistor connected to the word line is short-circuited between its base and emitter or emitter and collector. This arrangement permits error correction insofar as further dummy lines are provided for the word lines and a further dummy output line is provided for the output lines of the OR array. In these output lines, a random binary pattern may be stored, as the cross point between the dummy line and the PLA line connected thereto is controlled by one storage element. The storage elements in the dummy lines are combined in the form of a shift register. Thus, this known arrangement only permits detecting very specific errors, i. As the cross points in the arrays of this PLA, which realize the desired binary functions, are not provided with storage elements, this arrangement cannot utilize the advantages of dynamically personalizing a PLA with storage elements at the cross points. Therefore, this known arrangement is not suitable for systematically and completely testing the PLA. The invention has the advantage that it permits localizing and correcting an error in the PLA during its operation in a data processing system, since the whole test and load process requires only a relatively short time. This is achieved by fully utilizing the advantages inherent in the use of storage flip-flop circuits at the cross points. The novel design of the PLA, which is marked by the general use of storage flip-flop circuits both for the dummy lines and the remaining lines of the PLA, as well as by special set and reset lines, by means of which, for example, the OR array can also be set or reset columnwise, considerably shortens the test method and permits errors to be localized more accurately. A

single interrupted transistor, for example, may be localized by means of its row and column coordinates. The advantages offered by the invention are also obtainable in those cases where the product lines connecting the two arrays and the storage flip-flop circuits are not individually accessible from the outside. As the arrangement in accordance with the invention is used only for a short time in each case, it may be provided for many programmable logic arrays in common. Automatic testing and automatic error correction eliminate personnel for repairing the system. As loading and testing require only a short time, these operations may be effected prior to switching on the system in the morning. The whole test and load process, for example, takes only 1 msec. The invention will be described in detail below by means of the accompanying drawings showing one embodiment. A read-only storage 1 stores the micro instructions which are loaded into an instruction register 2. In each micro instruction a parity bit P is stored which is also loaded into register 2. The coded operation signals of the micro instructions are fed to the input lines of the PLA 3. At the output of the array 3 function lines are provided which are connected to particular control inputs in the data processing system. Depending upon the micro instruction, signals are received on particular function lines, which in a manner not shown here are used to control the system. The function lines are connected to the inputs of a parity circuit 4 generating a further parity bit from the binary 1-values received, which in the EXCLUSIVE-OR gate 5 is compared with the parity bit in the respective micro instruction. Normally, the two bits match. If they do not, an error signal CH is generated at the output of the EXCLUSIVE-OR gate 5, which in most cases is indicative of the programmable logic array 3 operating erroneously, as the remaining means are mostly far less errorprone and can also be tested in an entirely different manner. Thus, with the aid of the means shown in FIG. The output lines of the array 6 correspond to the columns of the array at the cross points of the row and column lines. There is, for example, one transistor, as will be explained further on by means of FIGS. Along the columns of the array shift registers are arranged which have as many stages as there are input or row lines. Each shift register stage controls its associated transistor. Thus, the logic state of the AND array is defined by the pattern stored in the shift registers. The inputs to the respective shift registers are designated as 10, 12, 14, 16 and 18 in FIG. If the left-most column of the AND array 6 is designated as the zeroth column and the top-most row line of the array as the zeroth row or line, the stages of the individual shift registers, which are arranged in the form of an array, can be designated as LU np in accordance with the indices for row line n and column p. The shift register belonging to the zeroth column thus comprises the stages LU 00 to LU Analogously to array 6, transistors are also provided at the cross points of the OR array 7. As in array 6, these transistors are controlled by stages of shift registers. In contrast to AND array 6 in which the shift registers are formed along columns, the shift registers of the OR array 7 are arranged along the row lines. The input signal for this shift register is applied to terminal The shift registers associated with the further lines 1 to 3 of the OR array 7 can be loaded at the terminals 32 to Notice that in the AND array 6, all stages of a particular shift register are set or reset together, but in the OR array 7 only stages of the same order column of the various shift registers are set or reset together. Also in accordance with the invention, there is an OR dummy line OEL which is associated with the zeroth line of the array 7. The former line is provided in addition to the output lines AL1 to AL3 of the OR array 7, which are arranged in the form of an array line. In the case of a single error in the PLA, the two dummy lines serve to switch off the affected cross points, replacing them by dummy crosspoints. The logic arrangement located at the output of the OR array 7, which will be described further on, permits replacing one of the output lines AL1 to AL3 by the dummy line OEL. The transistor of a cross point can no longer be switched on or off, and a line is either interrupted or short-circuited. In the following description these states are defined in such a manner that the respective transistor or line is stuck at "0" or "1". For explaining more clearly the types of errors encountered and the test method for detecting them, two possible implementations of the cross points of the AND array 6 and the OR array 7 will be described below. By means of the cross points in matrix 6, for a particular column, an AND condition is to be realized for

particular logic values on the input lines. If a signal is to be produced, for example, on product line PL4 when a 1 logic level signal is applied both to the input line 21 and the input line 23, the two flip-flop circuits LU 04 and LU 34 have to be set in the respective column. When the AND function is obtained, the transistor controlled by the flip-flop circuit LU 04 as well as the transistor controlled by the flip-flop circuit LU 34 not becoming conductive, thus causing the product line PL4 to assume plus potential. This is only possible if all the remaining transistors of this column are already non-conductive. If only one of the remaining transistors were conductive, the product line PL4 could not assume plus potential. In accordance with this, the conditions shown in FIG. Thus, the transistors T must be open non-conductive in any case if the flip-flop circuit is not set first and third line. If the flip-flop circuit is set, transistor T must become conductive when a binary zero is available at the input row line and it must be opened when a binary one is available at the input second and fourth line, respectively, of FIG. The logic conditions shown in FIG. In this circuit the collector of the npn transistor T is connected to the product line PL, while the emitter of said transistor is connected to ground GND. The input line is connected to an inverting input of an AND gate 58, while the other input of said AND gate is connected to the true output Q of the flip-flop circuit LU The output of the AND gate 58 addresses the base of the transistor T. If both input conditions of the AND gate 58 are fulfilled, the output rises, i. The OR function is fulfilled by one of the transistors T connected to a particular output line AL becoming conductive. In accordance with this table, transistor T must become conductive if the flip-flop circuit LO is set and a 1-value is received on the product line PL. If the flip-flop circuit is not set lines 1 and 3 in FIG. In accordance with the potential conditions specified, the table for erroneous output lines AL or erroneous transistors T is obtained in accordance with FIG. This routine permits identifying the erroneous or faulty transistor or line and permits restoring the operation of the PLA by means of the dummy line UEL or the dummy line OEL.

DOWNLOAD PDF SECOND EUROMICRO SYMPOSIUM ON MICROPROCESSING AND MICROPROGRAMMING, OCTOBER 12-14, 1976, VENICE

Chapter 5 : Historic Gravestone Art Of Charleston South Carolina Download tags:Cahans Exodus. A

Rodnay Zaks Explained. Rodnay Zaks (born 10 February , Paris) is a French-born American author of many books on computer programming, including the seminal Programming the Z80 and Programming the

Hartenstein, Adres Keevallik Ed. Performance Evaluation in Xputer-based Accelerators; Proc. Co-Design and High Performance Computing: Custom Computing Machines vs. Application-specific Microprocessor Design Methodologies: Technology and Practice, Wollongong, Australia, pp. A Reconfigurable Arithmetic Datapath Architecture: A Novel Computational Paradigm: Weber second best paper award: MOM - a partly custom-design architecture compared to standard hardware, Proc. MOM - Map oriented Machine; in: Hardware Accelerators, Adam Hilger, R. Rammig, Paderborn, Nov. Co, Amsterdam R. Carlstedt , Preprints of the 5th Intl. The Universal Bus Considered Harmful; in: Konzepte der Mikroprogrammierung; in: CAD-Hilfsmittel hoeherer methodologischer Ebenen: Symposium on Microcomputers, Budapest, Hungary, Okt. The use of the block diagram language ABL; in: Piloty Proceedings of the Intl. Die vierte industrielle Revolution; Techniken der Zukunft 3, Nov. Zukunftsaspekte der Informationstechnik; in: Techniken der Zukunft, Bd. Report on Interconnect Modelling: Ist die Informatik eine Ingenieur-Wissenschaft? Synthese endlicher Automaten zur Musterklassifikation und Musterverarbeitung; Kap. Anwendung abstrakter Automaten in der Zeichenerkennung; in: Europe, Cambridge, UK, July 24, Xputer - Novel High Performance Computers: Why a Second Machine paradigm? Principles, Architectures, Performance; Tutorial on Xputers: Hermstedt, Mannheim, Germany, Oktober Xputer: University of Vienna, Austria, May 4, Xputers:

Chapter 6 : Rodnay Zaks Explained

Lipp, "Array Logic" in "Second Euromicro Symposium on Microprocessing and Microprogramming", Oct. 12 to 14, , Venice, pp. 57 to 64, proposes in FIG. 2 the use of storage elements for controlling a cross point in a PLA.

Chapter 7 : Euromicro 82 - [PDF Document]

Microcomputer architectures: Third EUROMICRO Symposium on Microprocessing and Microprogramming, October , , Amsterdam: preprints by Euromicro Symposium on Microprocessing and Microprogramming (Book).

Chapter 8 : EPA2 - Memory addressing system - Google Patents

Charleston is the oldest and second-largest city in the state of South Carolina. By the 19th century, Charleston was home to the largest and wealthiest Jewish community in North America. The Separate Baptist Movement - The Story Of The Early Beginnings Of The Fairforest Baptist Church In Union District FIRST BAPTIST CHURCH IN UPPER SOUTH.

Chapter 9 : Rodnay Zaks - Wikipedia

The Concept of Knowledge in Islam and Its Implications for Education in a Developing Country.