String, procedure and macros in microprocessor

Strings In Microprocessor

In order to understand strings, one has to keep in mind that a string is made up of an array of characters. The string data type is an in-built data type that is an array of 256 characters (type string= parked array of chaege). When stored in memory, the processor should know where the string starts and where it finishes. In order to know where the string finishes, in Pascal, the 0th element of a string is defined as the length of the string. So, if you try to access character 0 of a string, the number of characters stored in that array is returned, thus letting the processor to know where the string finishes.

The power arc behavior on HV insulator strings is studied with regard to both the testing procedures and the design of guard devices. More precisely, the paper discusses

the problems of firing the arc with an impulse or a fuse wire and the importance of the symmetry conditions of the supply and the return circuit in order to obtain reproducible and representative tests. The consequences of such testing procedures in the design of HV transmission lines are shown in some typical cases, that is, for vertical and for V- insulator strings.

A string-oriented operating system for Intel-8080-based microcomputers is described. The system consists of a hierarchy of virtual machines. The lowest level virtual machines extend the instruction set of the 8080 to include additional 16-bit arithmetic and logical instructions, new data types, and operators. The data types include strings and string operators derived from the SNOBOL programming language. A table data type is constructed from strings, and table-manipulation operators are provided. A bit-map data type

and associated operators are also included. An Input/Output Control System (IOCS) support device-independent IO to multiple devices and diskette files. File name aliases permit many logical IO streams to be dynamically mapped onto a restricted set of physical IO units. Pseudo device handlers expand the capabilities of IO devices and are transparent to application programs. Distributed command decoders interpret IO command strings. Once communication is established with a logical device, a low-overhead IO Vector mechanism may be used for further access. A keyboard monitor provides interactive debugging facilities to application programmers. System resource allocation is implementation dependent and is not embedded in the system nucleus. Multiple implementations over a range of system sizes have demonstrated the utility and adaptability of WIZARD.

Apparatus and methods for testing a microprocessor chip using dedicated scan strings

A test apparatus and method for design verification of at least one microprocessor chip includes a compatible Joint Task Action Group (JTAG) terminal for access to a plurality of computer functional units contained in the chip. A test input terminal included in the JTAG terminal receives a scan string, the string being coupled to each computer functional unit through a first multiplexer. The scan input string is separated by the JTAG terminal under program control into a series of dedicated scan strings, each dedicated scan string being supplied to a selected functional unit through the first multiplexer. Each functional unit includes start and stop scan clocks for testing the functional under program control using the dedicated scan train for the functional unit. A test output terminal included in the JTAG

terminal is coupled to each functional unit through a second multiplexer. The test results of the dedicated scan string under control of the scan clock are supplied to the output terminal through the second multiplexer. The compatible JTAG terminal includes further elements for controlling the scan clocks to select a targeted functional unit for testing purposes while the scan strings for non-targeted functional units remain in an inactive state.

Macros In Microprocessor

A macro is a set of tasks combined together so that you can run or replay the entire task together with a single command. Macros are a powerful productivity tool. With macros you can perform long or boring tasks just by a single click.

If you think you are doing the same task again and again and it is frustrating and wasting your time and energy, you are ready to use macros. Even if it is not getting on your nerve, using a macro is a smart and fun way of working.

A microprocessor with a macro-rom exhibits reduced latency time and greater flexibility by including both a macro-rom queue and a main program queue. The arrangement eliminates the undesirable latency associated with fetching program as part of a return sequence from a macro-rom instruction. Also, the arrangement allows parameters to be extracted from the main program queue as the macrosequence is executing from the macro-roms program queue.

Field Of The Invention :

The integrated chip greatly improved the use for transistors, but it could only do what it was originally programmed to do. It couldn’t change programs, and it certainly couldn’t remember anything.

This invention relates to microprocessor organizations and more particularly to such an organization including a macro-rom.

Background Of The Invention :

A microprocessor includes a datapath portion and a control portion. Data and addresses are manipulated in the datapath portion. The control portion is operative to decode instructions in a program into a form suitable for controlling that manipulation. Programs typically are stored in a main memory external to the chip and include sequences of instructions and data at specified addresses in the memory.

The control portion of the microprocessor conveniently comprises a programmable logic array (PLA) for decoding instructions from main memory as well as auxiliary logic circuitry for applying decoded instructions to the datapath. A PLA includes an input register and an output register each having a set of latches. Instructions from main memory are applied to the latches of the input register typically during a first phase of each clock cycle of operation. During a second phase of each cycle, the latches of the output register are set to provide the binary code for controlling the datapath for the next subsequent cycle of operation. An instruction applied to the input register is called an op-code, and the output of the PLA (output register) is called a line of microcode. Each such line of microcode determines the “ state” of the microprocessor for the instant cycle of operation.

A PLA is characterized by feedback loops between the output register and the input register. These feedback loops carry binary data back to the input register to modify some bits of the input to the PLA in a manner to generate a sequence of related states. A PLA is able, thus, to generate a sequence of related microcode lines in response to each of one or more instructions in the program.

As is most often the case, data located at more than a single address in the main memory are required in order for even a single instruction to produce useful results. These data must be accessed and moved to (“ fetched” from main memory) on-chip registers in the datapath under the control of consecutive microcde lines in response to the single instruction. It typically takes a number of clock cycles to accomplish this movement of data even in response to a single instruction.

The requisite number of clock cycles for such movement is reduced if the microprocessor includes an on-chip queue in which the instructions and data for a portion of a program can be stored. If this portion of the program is “ prefetched” (i. e., fetched during earlier cycles) and stored in an on-chip queue in consecutive locations in the queue, the program can then be executed without wasting extra cycle time to access data stored in the main memory. Instead, the requisite instructions and data, when required, are obtained in a single cycle from the first location in the queue. Instructions in the queue are then applied to the input register of the PLA, and data in the queue are applied to elements of the datapath. Limitations imposed upon the speed of microprocessor operation by the bandwidth of the input/output (I/O) bus which carries instructions from main memory are thus reduced in

microprocessors which include such a program queue into which such prefetched instructions and data are stored temporarily.

A macro-rom is used to store on-chip, frequently-used programs called “ routines”. Such routines are often called for in the execution of certain instructions called “ macro-instructions.” A macro-rom is a word organized, on-chip, read-only-memory (ROM) operative to generate an ouput sequence of binary codes (coded words) in response to a corresponding sequence of input codes. The input codes are applied to the macro-rom from an on-chip register controlled by the output register of the PLA.

Operation of the macro-rom is initiated when a program in main memory calls for a macro-instruction to be applied to the input register of the PLA. The PLA responds to generate microcode, specified bits of which set specified latches of the output register of the PLA for configuring the datapath elements (i. e., the queue, counter, address register, . . .) to execute routines stored in the macro-rom and for activating the macro-rom as well. In turn, the macro-rom applies appropriate portions of the routine to the PLA input register. The routine is selected by the macro-instruction which specifies the addresses in the macro-rom at which the firt byte of the selected routine is stored.

Consecutive macro-rom outputs typically are not applied directly to the PLA because a macro-rom instruction is not necessarily aligned in a proper field for the input register of the PLA, and execution is slow due to the requirement of several clock cycles for accessing a macro-rom memory to obtain an instruction. Instead, the selected macro-rom program is also stored

in the queue. However, the selected routine cannot be stored in the queue without first erasing all unexecuted data then stored in the queue when the macro-rom is activated. The reason for this is that the queue is a sequential memory which can be loaded only from one end and read out only from the other. In the absence of erasing the unexecuted data, the routine from the macro-rom thus would not be located properly with respect to the unexecuted program already in the queue and would often occupy more space than would be available in the queue. Consequently, for proper operation, unexecuted program is erased and the queue is filled with a routine from the macro-rom.

Procedure In Microprocessor

The suboptimum detection procedure based on the weighting of partial decisions (WPD) was introduced as an improvement of one-bit-quantisation digital matched filtering, also known as binary matched filtering (BMF). The WPD is characterised by minimal additional hardware and software requirements but considerably better performance in comparison with BMF. A primary application of the WPD is the implementation of cost-effective medium-speed voice-band data medem receivers, but it can also be used in a number of other parametric and nonparametric detection problems. Formerly, the WPD was analysed only for binary transmission with an antipodal set of signalling waveforms. In this paper, the concept of the WPD is generalise and analysed theorectically for M-ary transmission with an arbitrary set of equal-energy signalling waveforms. Here, it is treated as the generalise procedure with BMF is its special case. The results of the performance analysis are provided, as well.

These Operating Procedures outline the orderly transaction of business of this committee.

For the development of standards, openness and due process must apply, which means that any individual with a direct and material interest has a right to participate by:

a) expressing a position and its basis,

b) having that position considered, and

c) appealing if adversely affected.

Due process allows for equity and fair play. In addition to openness, due process requires balance, i. e., the standards development process should have a balance of interests and shall not be dominated by any single interest category.

Refrences

1- www. macro-automation. htm

2- www. microstat. php. htm

3- www. answers. com

4- www. microprocessor. htm

5- www. micropinv. htm