1995/21558

DESIGN OF A REAL-TIME WIND TURBINE SIMULATOR USING A CUSTOM PARALLEL ARCHITECTURE

N95-27979

John A. Hoffman, Paragon Pacific, Inc.,
El Segundo, California

and

R. Gluck and S. Sridhar, TRW Space and Technology Group,
Redondo Beach, California

ABSTRACT

The design of a new parallel-processing digital simulator is described. The new simulator has been developed specifically for analysis of wind energy systems in real time. The new processor has been named: the Wind Energy System Time-domain simulator, version 3 (WEST-3).

Like previous WEST versions, WEST 3 performs many computations in parallel. The modules in WEST 3 are pure digital processors, however. These digital processors can be programmed individually and operated in concert to achieve real-time simulation of wind turbine systems. Because of this programmability, WEST 3 is very much more flexible and general than its two predecessors.

The design features of WEST-3 are described to show how the system produces high-speed solutions of nonlinear time-domain equations. WEST 3 has two very fast Computational Units (CUs) that use minicomputer technology plus special architectural features that make them many times faster than a microcomputer. These CU units are needed to perform the complex computations associated with the wind turbine rotor system in real time. The parallel architecture of the CU causes several tasks to be done in each cycle, including an IO operation and the combination of a multiply, add, and store.

The WEST 3 simulator can be expanded at any time for additional computational power. This is possible because the computational units are interfaced to each other and to other portions of the simulation using special serial buses. These buses can be "patched" together in essentially any configuration (in a manner very similar to the programming methods used in analog computation) to balance the input/output requirements. CUs can be added in any number to share a given computational load. This flexible bus feature is very different from many other parallel processors which usually have a throughput limit because of rigid bus architecture.

INTRODUCTION

The need for real-time simulation arises during many phases of the development and operation of systems with complex dynamical characteristics. The wind energy system represents such a complex system with many components whose dynamical characteristics couple and interact.

Simulation tools are indicated for such systems, to-

- support the initial design efforts for new systems,
- analyze the performance of system designs under the many variations in environment they will experience during their life cycles,
- evaluate failure modes and effects.

The mathematical models for a high-fidelity wind turbine simulation are very complex, especially if the significant aerodynamic nonlinearities are included. This complexity requires a powerful digital processor if real-time solution speeds are to be attained. Recent experiences with the control of wind energy systems, for example, has again underscored the need for good simulation tools to support the design and evaluations of control systems before these are placed in the actual operational environment.

Most past simulations have been nonreal-time, due largely to the inadequate computational throughputs of available computers in solving the complicated dynamical math models associated with wind energy systems. These slower simulations have provided valuable design support, but have been very limited in their use because of their cost and complexity.

The real time facility is very desirable if the simulation is to produce significant output within reasonable periods of time and at modest cost. Such fast simulations can provide invaluable support for the design process. The real-time capability is essential if the simulator is to be used in a mixed mode, where actual field hardware is validated in a simulation environment before being integrated into the final wind energy system. This validation process can reduce the risk of operating with new control systems, for example, by proving the systems in a simulation environment before risking operation with a real system.

The Wind Energy System Time-domain (WEST) series of simulation systems has been developed over the past few years to meet the needs for powerful real-time simulation tools to support future development of wind systems. The next section of this paper presents a background description of the WEST system developments that have preceded the current development of the WEST-3 article.

PRECEDING PAGE BLANK NOT FILMED

A description of the WEST-3 hardware system concept follows the background presentation. The software tools available to support development of programs to run on WEST-3 are then described. Finally, some of the plans for future refinement of WEST-3 are presented.

BACKGROUND

The orginal WEST unit was derived from a rotorcraft simulation system. The equations were modified for the wind turbine, and additional mathematical models were added for components not found in rotorcraft: the wind, tower, wind turbine controls, power generating equipment and the power train connecting the rotor to the generation system. The WEST-1 article is a hybrid system incorporating both digital and analog hardware. Because of the fast throughput needs, the hardware performing the mathematical calculations is substantially analog, with digital systems in place to act as executive controllers over the analog processes.

The WEST-2 system is also hybrid. It received expanded and refined mathematical models that were added to the baseline models of WEST-1. New features in WEST-2 included statistical math models for the wind and a programmable general-purpose hybrid subsystem for use in designing and evaluating new systems such as wind turbine controls.

The initial effort toward development of WEST-3 was directed toward the addition of specific additional computational facilities. Needed were math models for the rotor gimbal (teetering) system, higher-frequency blade aeroelastic degrees of freedom, expanded numbers of modes for the tower and supports and more refined models for the wind environment. The initial plans called for the refinement of WEST-2 to add these and some other needed improvements.

Before work in updating WEST-2 began, a new technology was emerging from a project supported by TRW Incorporated. This effort was directed toward simulation of large spacecraft structures, and produced an all-digital parallel processing concept promising orders- of- magnitude increases in digital system computational throughput. In the face of this new and promising technology, the decision was made to redirect the WEST-2 refinement effort toward a totally new system, WEST-3.

The generic technology incorporated in WEST-3 has been given the name: "Custom-Architectured Parallel Processing System", CAPPS. The CAPPS concept removes many of the objections raised in the past regarding architectures such as the WEST-1 and -2 systems. The primary limitation in these systems is seen in their analog implementations which are "hardwired" and which possess the limitations in significant figure accuracy associated with analog embodiments. The primary advantage of the analog archicture is, of course, its parallelism; this feature gives analog systems significant speed advantages over digital processors.

The CAPPS concept essentially borrows from the analog technology its best feature: its parallelism and therefore its speed. CAPPS also retains the primary advantages of the digital technology: programmability and accuracy. A special interconnecting concept was developed for CAPPS allowing the computational units (individual digital processors) to be configured into any overall system arrangement tailored to the specific application.

The next section of this paper describes the specific hardware architecture of the CAPPS in detail.

WEST-3 represents the most advanced simulation technology available in the WEST series of systems. The earlier units still have considerable utility; these can provide many of the functions needed in supporting future wind energy system simulation programs. The WEST-3 is a much more general system, however, and therefore has many application areas other than wind system simulation. CAPPS units with hundreds of computational units are envisaged, which promise simulation speeds orders of magnitudes faster than those currently available even with the fastest computers ever constructed.

THE CAPPS ARCHITECTURE

Figure 1 is a logical block diagram of the CAPPS concept. A series of "Computational Units" (CUs) are interfaced with a patch panel system via a series of serial "Input/ Output (IO)" data ports. These ports can be configured in any random manner connecting the CUs together in optimum configurations depending on the problem being solved. Each port is represented physically in the system by a single wire.

Each CU is in itself a very high speed digital computer. The current CU design has a 270 nanosecond (about 1/4 microsecond) instruction execution time. A complete series of operations is performed in a single

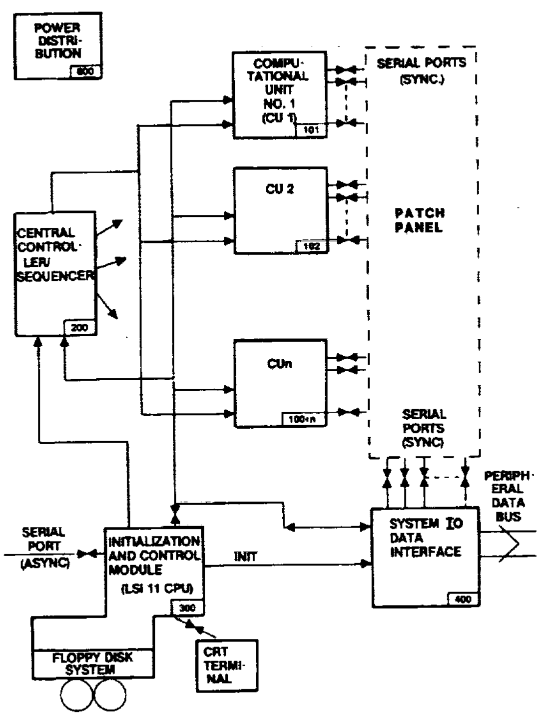

FIGURE 1 CAPPS SYSTEM LOGIC DIAGRAM

instruction, including:

- -instruction fetch
- -instruction decode
- -two operand fetches
- -a result store
- -a full word multiply of two arbitrary operands
- -an arithmetic logic operation on two operands, including add, subtract, shift or boolean operation (OR, AND, Exclusive OR, etc.)
- -a full word IO operation (either input or output under software control)

The performance of all of these operations in one instruction cycle makes the design of a CU a very fast processor, even in singular operation.

The IO operation performed during each instruction cycle is automatic and enables as many CUs as desired to be connected together in any random configuration for parallel processing operations. This feature makes the overall CAPPS design truly unique; there are no longer any real limits placed on the time frame or speed associated with the solution of any technical problem. The number of CUs can be increased without bound until adequate computational resources are available to perform the problem at hand.

Figure 2 presents a more detailed definition of the CU architecture. Note that separate instruction and processing memories (RAMs) are incorporated so that a new instruction (and all associated operand addresses) can be fetched and decoded while the last instruction is being executed. Additionally, the CU incorporates a parallel multiplier which is a dedicated arithmetic subsystem that multiplies two operands together in about 100ns. The rest of the Arithmetic Logic Unit (ALU) is also depicted by Figure 2, along with the accumulator and loop back to processing RAM for storing the resulting calculation.

The architecture presented by Figure 2 is often referred to as a "pipelined" system, in that multiple computational stages are connected in a string and perform "added value" computations on a data flow as it moves down the imaginary computational pipeline.

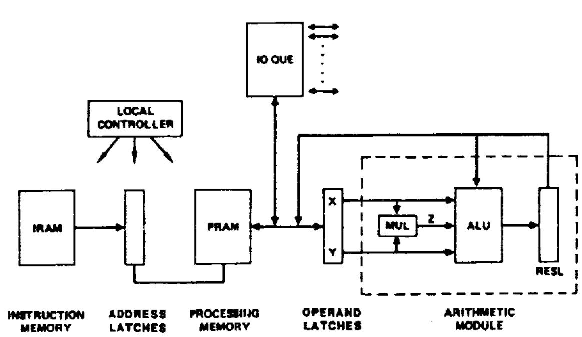

FIGURE 2 COMPUTATIONAL UNIT OF THE CAPPS

Figure 2 also shows the IO Que; this is the subsystem that enables the interfacing of the CUs in any random configuration, a capability that is unique to the CAPPS architecture. The IO Que in the present design incorporates 16 IO ports, although this number could easily be increased or decreased for future specialized designs. The Que has 32 registers, each with a 16-bit capacity. The Que is separated into two "banks" (say bank A and bank B), each with 16 registers. Figure 3 shows the register arrangement in the IO Que design.

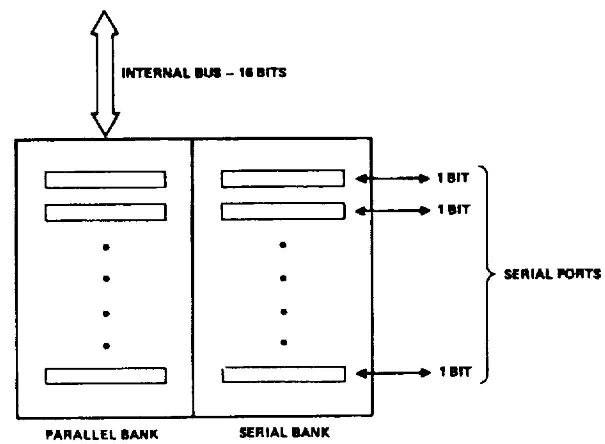

FIGURE 3 IO QUE SYSTEM OF A COMPUTATIONAL UNIT

During operations, one bank is in parallel mode while the other is in serial mode. The parallel bank experiences parallel data operations between its registers and the CU processing memory. A full word is moved into or out from a register in the parallel bank during each instruction execution.

No parallel accesses occur in the serial bank, because this bank performs as a series of 16 shift registers during operations. During each CU instruction cycle, each of the 16 registers shifts in or shifts out one bit from or to the serial port (single wire) connected to it.

The IO Que "toggles" every 16 instructions. At the end of each group of 16 instructions, all 16 registers in the parallel bank have been interfaced to processing RAM (via a procedure often called DMA for Direct Memory Access) and all bits have been shifted into or out from all of the registers in the serial bank. After the set of 16 cycles, the serial bank is switched to parallel mode and the parallel bank is toggled to serial mode. The process continues indefinitely, as long as the CUs are in operation.

Figure 1 shows the "Central Controller/ Sequencer" (CCS) subsystem which synchronizes the operations of all the CUs. The entire CAPPS unit has only one clock (in the CCS) which clocks all of the CUs in phase. In this way, all CUs receive or transmit data bits over their serial ports synchronized together. Each unit has an internal strobe that advises when to send or receive bits.

Figure 4 shows an example of how a series of CUs might be connected together to solve a particular problem. Note that there are no constraints on the arrangements of the buses or ports among the CUs; also, of course, there is no limit on the number of CUs that can be connected to share in the execution of a problem.

Because the CAPPS has been designed specifically for the simulation of large numbers of time-domain equations possessing significant nonlinearities, the instructions in the CU have been heavily biased to perform operations consistent with these types of equations. For example, the CU instruction has a three operand format. Two operands are fetched, processed (including multiplied), added or subtracted from an accumulating sum (if desired) and then stored. Most processors have two operand instructions, so more instructions are required to do operations such as sums of products.

In simulations of structural dynamics, controls and many other applications, the equations appear substantially as sums of products. This is the primary form of processing associated with math models in matrix or tensor form. For these types of equations, the three-operand instruction is significantly more powerful than the two-operand systems. Hence, the computational throughput is enhanced accordingly, for these types of problems.

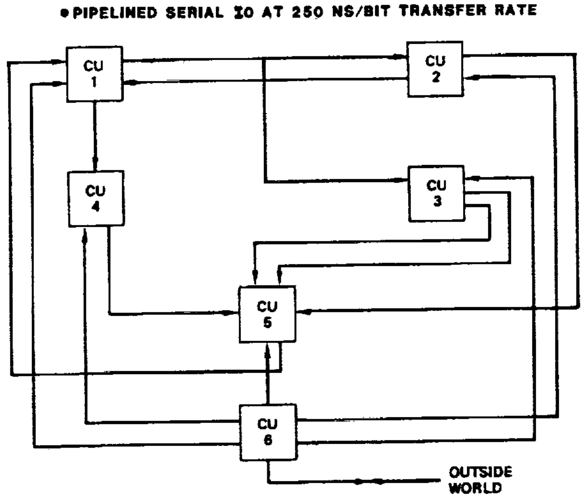

FIGURE 4 TYPICAL SIMULATION CONFIGURATION

In this and other ways, the CAPPS architecture has been biased specifically for the application; the system therefore achieves much faster speeds than processors that are designed for more general applications.

Because the CU architecture is accessable (i.e., it is not built into a chip where it cannot be changed to match special needs), it can always be enhanced in special ways for special problems. For example, additional memory banks may be added for parallel fetches if large tables of data are to be processed. The CUs can be altered at any time to match the needs of the application.

Figure 1 also shows the Initialization and Control Module (ICM). The ICM is responsible for loading data into the CUs at initialization time, usually from the disks also depicted by Figure 1. The ICM loads code, data tables, flags, etc. into the processors. It then clears the program counters in all CUs at once, and then instigates parallel execution.

The WEST-3 ICM incorporates a 16-bit microcomputer system, the Digital Equipment Corporation (DEC) LSI 11. This processor was chosen because of the very large body of proven software that exists for this system, including a reliable Fortran compiler.

Although the ICM performs many sophisticated tasks associated with CU management, it is itself too slow to contribute significantly to the actual calculations made by the system.

The CUs incorporate built-in logic analyzers which allow the ICM to "single step" through programs and read the data on the many internal CU buses. The data is displayed on a terminal or printed to aid programmers in developing codes for CU execution.

At initialization time, the ICM loads IO tables into all of the CUs. These tables tell each CU which parameters are to be communicated over which IO ports. Whether the operations are to be input or output is also specified for each port. Data can be communicated in blocks, so each port can transmit blocks of parametric data of any size. A single port can connect to many CUs and transmit the same blocks to all of them. Additionally, ports can be connected together so that a number of CUs transmit data over them in time- multiplexed fashion. The ports can be shorted without damage, and can drive lines at least 10 feet long.

The port data rate is 4 megabits per port. All ports on a CU communicate with the outside world at 64 megabit rates.

Figure 1 also shows a "System IO Data Interface (SIDI)". This interface will change from unit to unit depending on the needs to connect the CAPPS to other computer devices. Large mainframe computers, disk drives and graphics terminals are candidates for the SIDI peripheral data bus. The SIDI will enable the communications of massive amounts of data in various formats. The SIDI can also incorporate analog interfaces, digital-to-synchro converters or other special devices depending on the application. The SIDI incorporated in WEST-3 is a pure analog interface at this time. It converts internal digital signals to analog for up to 64 independent channels for display on standard devices such as strip-chart recorders and memory oscilloscopes. The WEST-3 SIDI also has provision for 64 channels of analog input information, so that the simulator can be connected to real wind turbine control hardware, wind measurement signals, etc., to act as a component within an overall simulation environment.

THE WEST-3 DEVELOPMENT PROCESS

WEST-3 was actually fabricated twice during the full development process. The first unit performed its calculations correctly but unreliably. Also, it was unable to attain full-speed operation.

There were a number of problems with this first CAPPS prototype, including:

-Logic Errors in the design, particularly in the controllers (e.g., the CCS and ICM interfaces). There were also some major items needed, however, (such as a special IO counter) that were not included in the first design.

-The system wiring computer program was not directed to place specific modules in specific places in the system when the first prototype was fabricated. Although this program attempts to optimize wire lengths, its built-in algorithms were simply inadequate. The result was excessively long buses which developed "cross-talk" problems (spurious communications between proximate lines in a bus due to electromagnetic and/or capacitative coupling at high frequencies).

-The grounding system which has worked acceptably in the past was inadequate for the high speed CU. It developed large transients which were able to falsely clock and clear registers in the system.

-Excessive delays appeared in the system due to the choice of (relatively slow) "LS" digital logic for implementation of the CUs.

-Unbalanced timing appeared, especially in a number of the control functions, due to excessive logic stages in certain critical signal paths, and due to unequal numbers of logic stages in areas requiring balance.

-Excessive noise on the bus between the ICM and the CUs caused errors in data loaded in the CUs at initialization time and data measured using the built-in logic analyzers.

Because of these many problems, the original WEST-3 was completely reconstructed. Major design changes were made to eliminate the problems observed with the first system. The changes made are summarized below:

- Special logic functions, especially the controllers, were isolated to single boards enabling convenient changing of the logic to correct errors, and more importantly, enabling "fine tuning" of the system timing to get maximum performance.

- Modules in the system were carefully located (the wire list program automatic-placement mode was preempted by designer location choices) to minimize bus lengths. Compromises were made favoring buses with critical timing over those with less stringent requirements.

- A new grounding system was developed and installed incorporating large gold-plated strips and mil spec connectors on the strips and boards to engage the grounds.

- The "LS" logic technology was discarded and new integrated circuits were purchased of the "F" TTL line (for "fast"). The F TTL technology is brand new. It features the low power consumption of LS, and is faster than "S" TTL. Indeed, F logic rivals the very fast ECL technology while enjoying significantly lower energy requirements and involving the much simpler TTL design rules.

- The control logic was carefully tailored and balanced using digital delay networks on critical timing modules. The networks enabled the adjusting of pulse timing in increments of 5 ns, to fine tune the system for maximum speed.

- A special ICM Interface using the latest (low noise) CMOS technology was incorporated to eliminate data communication errors between the ICM and CUs.

- Because of the very high speeds associated with the F logic, control lines in the CUs begin to behave as transmission lines. To avoid large pulses caused by reflected waves on these lines, they were terminated with resistors chosen to match their characteristic impedances. These terminators reduced the noise signatures on the control lines to acceptable levels.

- Much of the CCS pulse-shaping logic was moved from the CCS to local controller modules, thus reducing line lengths for critical timing signals. Now only two twisted-pair lines communicate the clock and a synchronizing signal between the CCS and the CUs.

As mentioned previously, the new system performed reliably and accurately at maximum speed.

THE ICM DEVELOPMENT

A number of difficulties were encountered with the ICM in reducing it to practice. A poorly taped printed circuit board packaging the LSI 11 processor developed crosstalk problems and had to be refabricated. Additionally, problems with the standard software available for the PDP1103 (particularly associated with the disk and system port handlers) required development of new handler programs and a special interrupt controller not originally planned for the ICM system.

These problems have now been solved so that the compatible ICM modules are now available for integration into the WEST 3 article. At the present time, the ICM resides in an enclosure separate from the WEST-3 computational units.

THE HARDWARE VALIDATION

The WEST-3 design was proven at the maximum anticipated speed of 270ns per instruction cycle. Two programs were developed and executed. The first was used with the internal logic analyzer to exercise each instruction in the full CU set and print the results. The results were examined to prove proper static operation of all elements of the system.

A looping function generation program which exercised most instructions in the set was then developed and executed. The generated traces, linear and parabolic sawtooth functions, were output through the analog SIDI system and displayed on a memory oscilloscope. These exercises proved correct dynamic operation of the CU at full speed.

THE TRW DEMONSTRATION

TRW Inc. has developed a demonstration code for the CAPPS unit using a benchmark math model which bears considerable simularity to the types of spacecraft dynamics problems they wish to solve at high speed. The benchmark problem is for a flexible whirling beam undergoing a despin maneuver in space.

The whirling beam benchmark was run on a single CU first, and then on both CUs in WEST-3, demonstrating full parallel operation. Figure 5 compares the theoretical response of the beam derived by a separate simulation to that achieved with WEST-3. Clearly the WEST-3 solution duplicates the baseline, proving the accuracy of WEST-3 and its ability to solve complicated dynamics problems using CUs wired in parallel.

The next section discusses the performance comparisons made for the CAPPS technology and other commercially available processors.

PERFORMANCE

In the course of searching for an advanced processor or concept to use for future simulations of spacecraft systems, TRW Inc. ran the whirling beam benchmark problem on a number of commercially available advanced processors. Figure 6 presents the results for the whirling beam problem. The performance of WEST-3 is seen to exceed the best of all the available processors with only two CUs. Note that the AD-10 performed the best of the commercially available processors,

COMPUTER	VENDOR	HOST COMPUTER	REALTIME/ CPU TIME
AD-10	APPLIED DYNAMICS INTL	PDP 11/34	58.48
CRAY 1	BOEING COMPUTER SERVICE	–	44.72
IBM 3081	IBM	–	27.78
IBM 3033	IBM	–	20.10
FPS 164	FLOATING POINT SYSTEMS, INC.	VAX 11/780	8.33
HEP H1000	DENELCOR, INC	–	4.08
CAPPS (2 PROCESSORS)			71.30

Figure 6 Simulation Results for the Whirling Flexible Beam Benchmark Problem.

better than the Cray super computer. Figure 7 presents another performance comparison among the latest available computers and a projected CAPPS configuration. Note especially the cost for running a typical spacecraft problem: $49,000 for a single case (100 seconds of real time in the simulation of a complex orbiter spacecraft) using the TRW IBM computer. The CAPPS cost is projected at only $23 for the same case, including reasonable acquisition, maintenance and operational costs.

Of course, Figure 7 does compare "service" processors with a CAPPS unit operating as a dedicated processor. An attempt was made to make a valid comparison in this case, however, by assessing all costs needed to run CAPPS in dedicated mode over its estimated life cycle. The weekly useable production time of CAPPS was conservatively estimated at only 18 hours to make the comparison.

Figures 6 and 7 reveal the very powerful promise of the CAPPS technology in general for future simulation needs. Applications other than simulation are, of course, indicated for this concept. Included among these are signal processing, control and automation tasks.

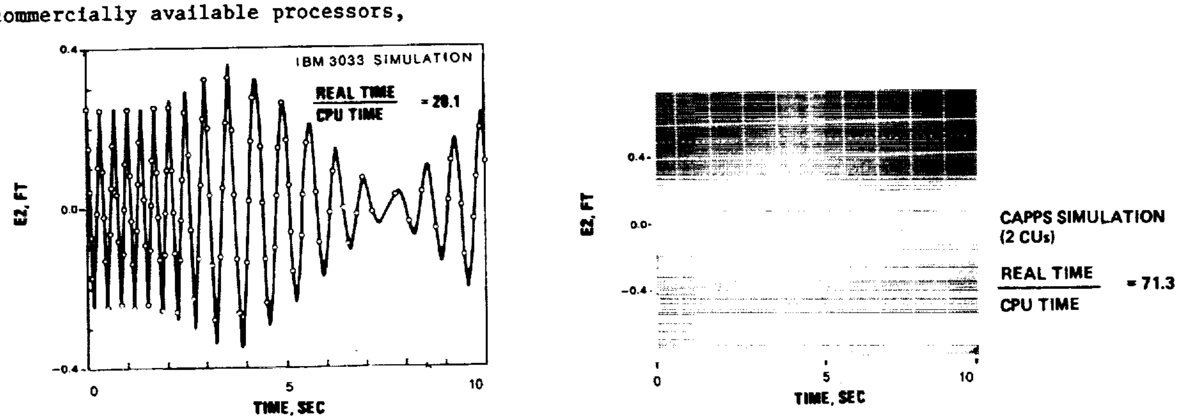

Figure 5 Demonstrator's Simulation of the Whirling Flexible Beam Problem

COMPUTER	VENDOR	CPU TIME/ REAL TIME	LENGTH OF RUN (CPU HRS)	COST OF RUN ($)	
CRAY 1S	BOEING COMPUTER SERVICE	183.2	5.09	8,433	MEASURED RESULTS
CYBER 205	CDC	410.0	11.38	33,920	
IBM 3081	IBM	792.3	22.01	39,329	
IBM 3033 (TRW)	IBM	1670.0	46.39	49,000	
CAPPS	(ESTIMATED RESULTS FOR 20 CU'S)	10.8	0.30	23	

FIGURE 7 SIMULATION RESULTS FOR THE ORBITER-RMS-PEP SPACECRAFT BENCH MARK PROBLEM

SOFTWARE SUPPORT

Figure 8 depicts the software modules currently available for supporting the development of code for CAPPS CUs. Two general capabilities are available: the system that produces code for actual CU residency, and the simulator.

TRW have developed a translator and assembler for the CU. The translator receives equations prepared in accordance with Fortran rules, and decomposes them into assembler form. Figure 8 shows examples of equations input to the translator, and the assembler syntax that emerges. The assembler than converts the results produced by the translator, and other assembly-language code supplied by the programmer (if any) into machine code. The machine code files are ready for direct loading into the CU for execution.

In addition to these development tools, considerable software has been developed for ICM residency which supports the operation of the WEST-3 and aids in validating programs prepared for the unit. ICM- resident codes load the CUs, provide for "single stepping" the CU through code for debugging purposes, and allow other convenient facilities such as "peeking" and "poking" CU memory to view intermediate computational results and to set up test scenarios used during program validation.

A special simulator program has also been developed, which will execute programs prepared for the CU on a PDP 11 microcomputer. The ICM can execute the simulator. The user can inbed debug code (such as debug printouts, breakpoint logic, etc.) into the programs read by the simulator. The simulator will duplicate the actions of the CU and produce intermediate results to aid the programmer in debugging programs developed for the CU.

The tools currently available for CU program development and debugging are not as sophisticated as a system incorporating a macro assembler, linker and Fortran compiler, but they do provide very significant services which approach the convenience of a full system. It is much simpler to develop CU code with the currently available tools than it is to develop programs in pure assembly language.

The next section discusses current limitations and future plans now being implemented for the CAPPS technology; continuing development of software tools is slated to be a major element in these future endeavors.

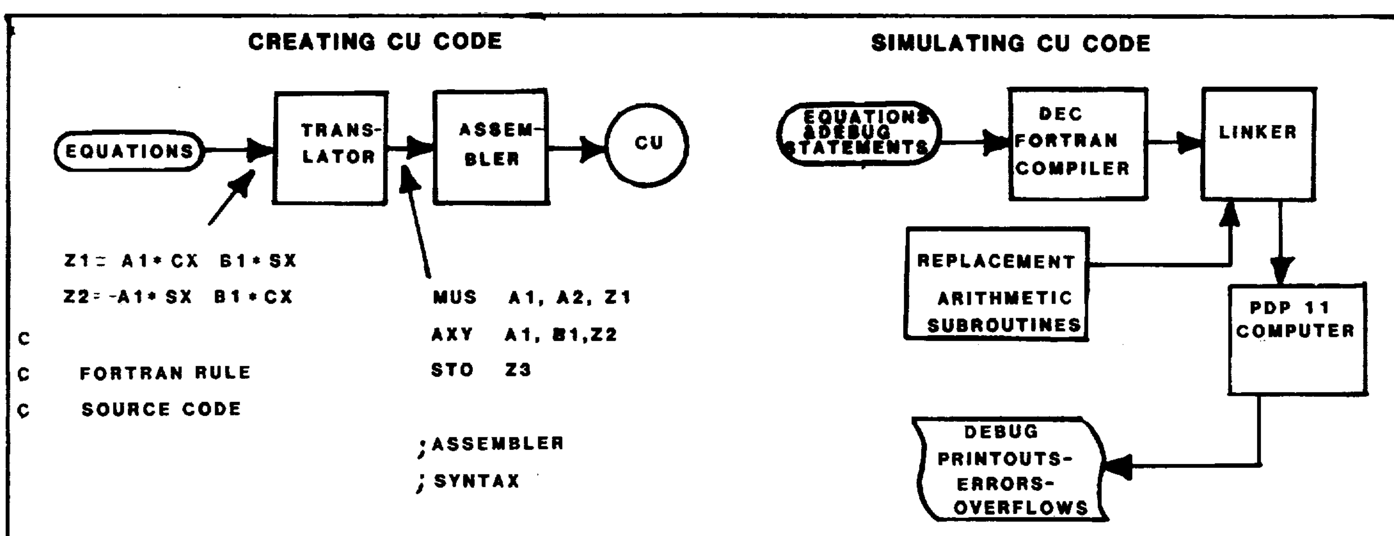

FIGURE 8 SOFTWARE TOOLS AVAILABLE FOR CAPPS/WEST-3 PROGRAM DEVELOPMENT

C-2

LIMITATIONS AND FUTURE PLANS

The current CAPPS CU architecture incorporates a 16 bit data bus and integer arithmetic. Equations prepared for the CU must therefore be scaled and cast as integer expressions. Additionally, the system does not currently have standard software modules available for it, the most pressingly needed being a Fortran compiler, linker and full macro assembler.

These limitations are not precluding the development of a complete mathematical model for the wind energy system application of CAPPS in the WEST-3 embodiment, but they do make the programming task more specialized and time consuming. Accordingly, a number of major developments are currently underway which will eliminate many of the limitations in the present WEST-3 system. More specifically, the following are now under development:

-A 32-bit hardware floating point processor that will operate at the same speed as the current CAPPS instruction cycle (270 ns per instruction).

-A hardware translator that will receive and execute code prepared initially in Fortran, macro assembly, or many other forms (Pascal, Basic, etc.). A translator will reside in each CU and act as an executive host processor. It will load directives for the CU so that the CU will perform the heavy processing tasks using library routines written for efficient CU operations. The resident translator will make the CU "transparent" to the user. The operating system planned for this new concept is the familiar RT-11 system offered by Digital Equipment Corporation, which has been available for years as a reliable and mature system for use in the PDP 11 line of minicomputers.

-More refined ICM resident software modules for use particularly in debugging programs.

-A winchester disk drive to augment the floppy disks now used with the ICM.

These and other refinements should be available for the CAPPS technology within the next year.

OPTIONAL ARCHITECTURES

Architectures other than the pure CAPPS arrangement of Figure 1 have been designed for the WEST-3 application. Figure 9 is the original configuration proposed for WEST-3. This system incorporates CAPPS subsystems, including the two devices called "Fast Processors" in Figure 1. These are essentially computational units that have been interfaced to an array of microcomputers to share the entire WEST-3 computational load.

The microcomputers are very slow compared to the CAPPS computational units, but they do have mature software support. In the original WEST-3 concept, the microcomputers were to execute math models associated with relatively slow elements of the system such as the tower, control system and wind models. The fast processors were to solve the complex equations associated with the rotor system. A "central data store" or shared memory is incorporated in the design to facilitate communication among the microcomputers, the ICM and the fast processors.

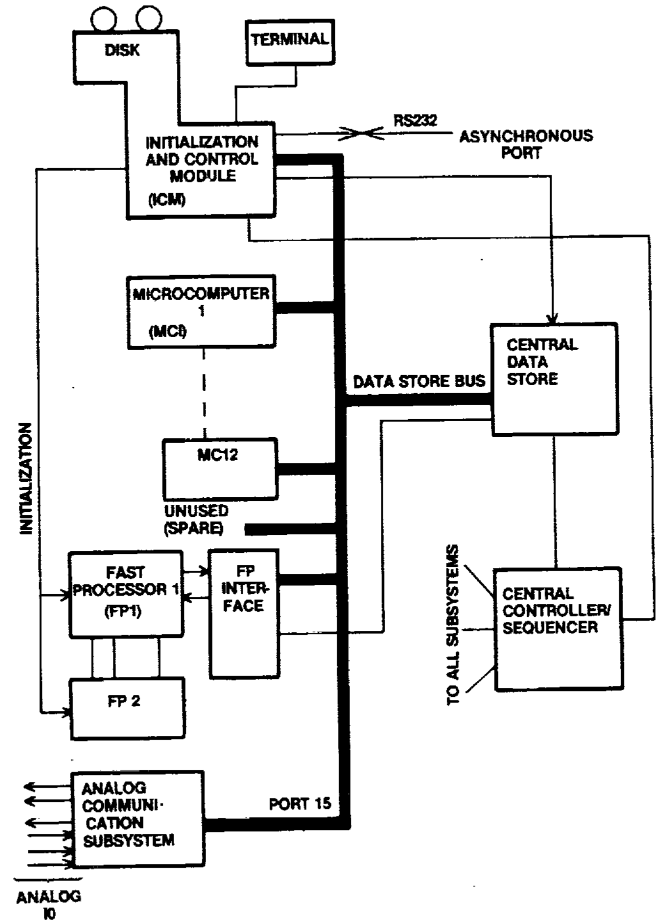

Figure 9 Optional WEST-3 Design

Since the inception of the original WEST-3 architecture of Figure 9, considerable effort has been expended toward developing software tools for programming the computational units. The availability of the translator and assembler modules discussed above have made programming of the computational units much more convenient and efficient than if pure assembly language programming were to be used. The desirability of the array of microcomputers has therefore diminished.

Figures 1 and 9 represent blends of processors that can be defined for an application depending on the specific characteristics of the equations being solved. The present thinking prefers larger numbers of computational units and fewer of the slower microcomputers for the WEST-3 application.

CLOSURE

The WEST-3 system has demonstrated the technical feasibility of the overall CAPPS concept. It affords the utilization of processors in massively parallel configurations, with a flexible bus architecture that can be tailored for the application. Since communication problems among individual processors has been a major limitation on parallel processors of the past, the CAPPS promises to offer significant increases in computational throughput largely because of its flexible configurational means.

A detailed mathematical model for the total wind energy system has been developed for WEST-3, and is currently being validated on the system. The models derive from those of WEST-1 and -2, but have been reformulated for digital solution. Additionally, significant additional modelling fidelity has been added, including a rotor gimbal, two more blade aeroelastic modes (for a total of three modes) and more generality in the control system, electrical power system and supporting structure models.

ACKNOWLEDGMENT

Many individuals have contributed to the development of the advanced simulation technology incorporated in WEST-3. The authors wish to express their appreciation to these people, and particularly, to Mr. Dan Weisz and Mr. Milton D. Campbell for their dedicated efforts in supporting this project.

WEST-3 has been developed under funding from the NASA Lewis Research Center, Cleveland Ohio, and the United States Department of Energy, Washington, D.C. The authors also wish to thank Mr. David Janetzke and Mr. Harold Neustadter of NASA Lewis for their considerable support and encouragement throughout the WEST-3 development program.