PRELIMINARY REPORT ON THE NOVIX 4000

C L Stephens, W P Watson

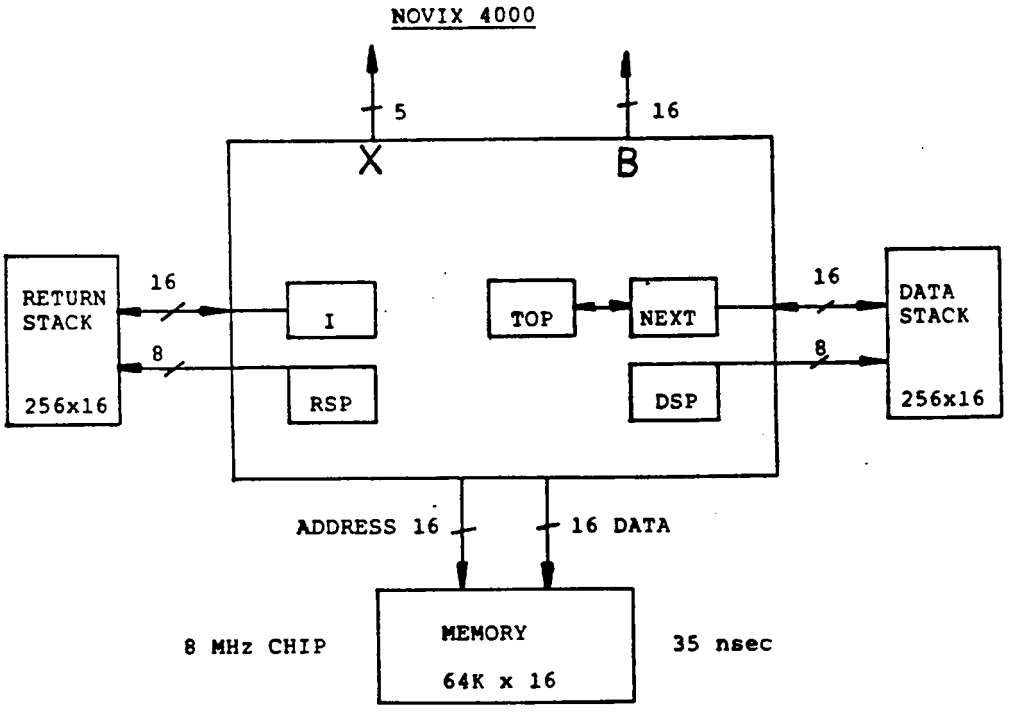
Computer Solutions Ltd
1 Goodmore Lane, Chertsey, Surrey KT16 9AP, England

The NOVIX 4000 is a true FORTH processor and is capable of in excess of 10 million FORTH instructions per second. It is implemented as a gate array.

This paper will introduce the architecture of the chip, its hardware configuration and the software support facility provided with it. Details of a number of benchmark programs will be given and these will be used to identify particular application areas for the chip.

Architecture

The NOVIX NC4000 is a gate array chip designed by a team led by Charles Moore. It implements FORTH as its "machine code" and as such, is the first true FORTH machine on a chip. The chip is a 124 pin conventional CMOS gate array that operates at one cycle every 125 nanoseconds (8 MHz). Its architecture is shown below and as can be seen, it implements separate buses for off chip Instruction and Data Memory, Data Stack, Return Stack and I/O.



The first point to make is that the machine is 16 bit word oriented and all its data buses are that width. The two stacks use separate off chip memory areas and 256 x 16 bit words are addressed in each stack. The Data Stack has its top two values held on the chip (N and T) in order to make the large number of FORTH words that take two parameters operate in single cycle instructions.

Subroutine Threaded Code

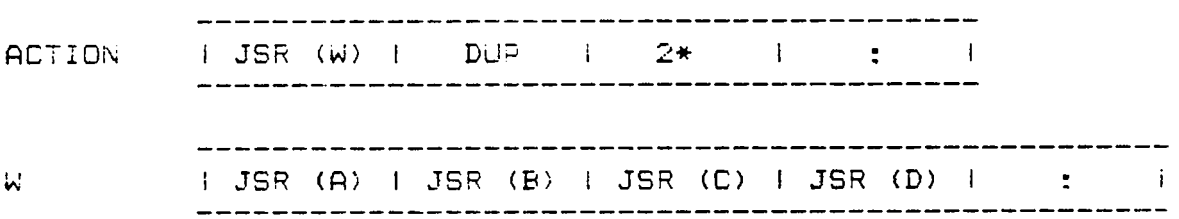
Rather than implementing conventional Indirect Threaded Code, the NOVIX uses Subroutine Threaded Code.

Hence a set of definitions such as

```
: ACTION W DUP 2* ;

: W A B C D ;
```

Will compile to (Brackets indicate an address) :-



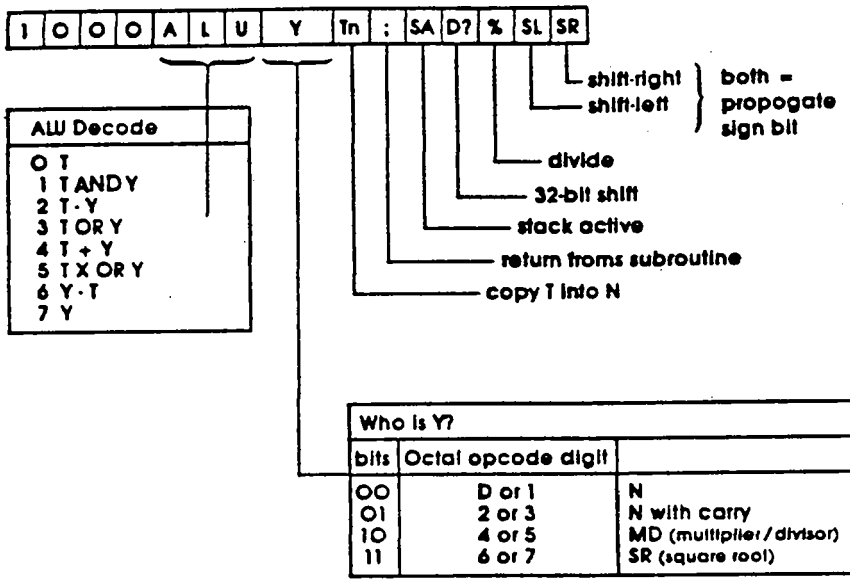
As this invocation of another module is so basic to a FORTH system, it is important that it should have the minimum overhead.

This is done by making the most significant bit of the opcode define either an "ordinary" instruction (1) or a subroutine call (0). If the top bit of the opcode is 0 then the remainder is the address to be jumped to. Hence, subroutine jumping takes only one cycle. While this limits FORTH definitions to only the lower half of the 64K word memory, it will be shown later that this is not thought to be a significant limitation. Of course, subroutine calling is only half the problem and an efficient call with a slow return would be no gain. The NOVIX can combine the semicolon action in with the majority of its instructions, by taking advantage of parallel access to the Return Stack. In this way, the single cycle add instruction can also include a return without increasing its execution time.

The Arithmetic and Logical Instruction

These opcodes all start 1OXXXX (octal) and may involve a shift, a Multiply/Divide Register (MD) and a Square Root Register (SR) as well as Top of Stack (T), Next on Stack (N) or Next on Stack + Carry (Nc).

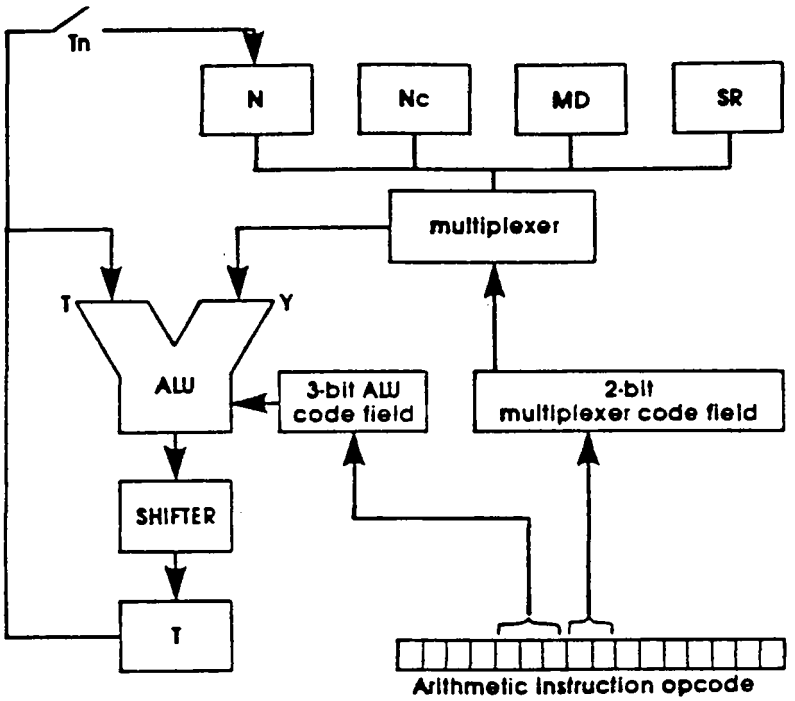
The structure of an Arithmetic Instruction is :-



While the FORTH programmer need not consider the layout of the ALU Instruction, (the FORTH Compiler will hide these details) it is important to look in some detail at the ALU in order to understand the speed and power of the NC4000. The important point to note is that the ALU instruction consists of a number of fields each of which control a section of the NC4000 internal components and all fields of which are designed to be executed in parallel. In this way, it is comparable to a horizontal bit-sliced architecture, but as each field represents a FORTH word, as many as five high level words may be executed in a single cycle (and incidentally compiled to a single 16 bit opcode). Because of this feature, the NC4000 is said to be a "High Level Externally Microcoded" machine. The resulting code compaction effect is why the 32K word limitation on program area is not thought to be a significant limitation.

These then give the following basic ALU instructions :-

Opcode (octal)	Instruction (Forth)	
100000	NOOP	No operation
100020	NIP	Pop N
107020	DROP	Pop N into T
100120	DUP	Push T into T
107120	OVER	Copy N into T while pushing T into N
107100	SWAP	Exchange N & T
104020	+	Add N to T and pop N



Opcode	Instruction	
104220	+c	Add N to T with carry and pop
106020	-	Subtract T from N and pop N
106220	-c	Subtract T from N with carry and pop N
103020	OR	Logically OR N into T and pop N
105020	XOR	Logically XOR N into T and pop N
101020	AND	Logically AND N into T and pop N
100001	2/	shift T right one bit with T15 into T15 and T14
100002	2*	Shift T left one bit with zero into T0
100003	0<	Propagate T15 into all bits of T
100011	D2/	shift the combination of N and T right one bit
100012	D2*	Shift the combination of N and T left one bit
104411	*'	Multiply step
102411	*-	Signed multiply step
102412	*F	Fractional multiply step
102416	/'	Divide step
102414	/"	Last divide step
102616	S'	Square root step

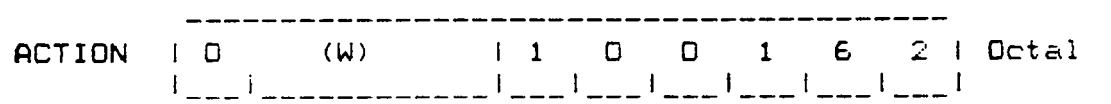
Some of the most useful multiple FORTH words that can be implemented in a single cycle are :-

```
OVER  +     ( a b -- a a+b )

SWAP  -     ( a b -- b-a )

+ 2/        ( a b -- a/2+b/2 )
```

and of course the ; can be combined with any of these so that the word ACTION now compiles to two sixteen bit words.



The multiply, divide and square root step operations require successive executions to perform a full high level operation, and there is a TIMES register in the NC4000 which if loaded with a value, will force the next instruction to be executed that many times to help in this.

```
16 bit   *    takes 20 cycles
16 bit   /    takes 25 cycles
16 bit  SQRT takes 27 cycles
```

Conditional Jumps

Three opcodes provide jump facilities

11XXXX	Conditional	IF / UNTIL / WHILE
12XXXX	Unconditional	ELSE / AGAIN / REPEAT
13XXXX	Loop	#LOOP

In these cases XXXX is an absolute address within the current 4K page. The #LOOP primitive and the #DO implement a count down to zero using only a single return stack value to hold the loops parameter.

Other Instructions

Other instructions include register access (I/O registers, R), etc), memory access (16 bit address and 21 bit using the X lines) and short literals (the five bit literal being part of the instruction). In all these cases, "external microcoding" is used to allow ALU operations to take place in the same cycle. Typical multiple operations that can take one instruction are :-

```
R)   SWAP  )R
@    SWAP  -
DUP @ SWAP nn + ( a -- m a+nn )
nn +
```

The only opcodes requiring more than one word are long literals in which the 16 bit literal follows the opcode.

I/O

I/O devices may, of course, be accessed via memory mapping if required, but two special bi-directional I/O buses are provided . the 5 bit X port (which may be used for extended memory addressing in some applications) and the 16 bit B port.

Each port allows individual bits to be specified as latched output, tri-state output or normal input. In the latter case, the value read from the port is the exclusive OR of the received value and a user set comparison register. So, a simple read of an input field (eg status) gives a truth value for direct testing. On output a mask register allows specified bits to be masked out to allow the bit field of the port to be written to independently. By using these, it is relatively straightforward to use say eight bits of the B port for bi-directional data, two for control and four for device addressing.

Hardware Implementation

The NC4000 achieves its high processing speed not just from being a high level language on silicon, but also from a number of hardware techniques employed in its design.

The most striking of these is the use of external dedicated hardware for the return and parameter stacks. Thus, there are three sets of buses coming out of the NC4000 - resulting in some 124 pins. However, while this appears at first glance to complicate matters, in practice the clock signals required for each bus are very straightforward so that it is in many ways easier to attach memory devices to the CPU than is the case with other 16 bit processors such as the 68000 or 80186 devices.

Internally, the NC4000 uses no microde - each instruction bit has a direct action on the internal logic within the device. As only some 4000 gates are used, this implies very concise internal logic - another factor in improving performance.

Externally, the fastest clock signal that can be used is 8MHz. With only some 62 ns per half cycle, any external memory decoding logic has to be fast - in practice, the combination of HCMOS parts with 35 ns RAM works well, and this is used on the current development system, the Beta Board which will be described in the next section. Generally, there is a read on the first half cycle and a write on the second half. Separate write strobe lines are provided for each memory area (parameter and return stacks, and main memory) and for the I/O port. Of course, it is not necessary to use 35 ns RAM which is expensive - 70 ns devices are more widely available. If the same clock transition times can be maintained, then doubling the memory

access time may not necessarily mean doubling the cycle time: a reduced clock of only 6 MHz may be possible.

The NC4000A is implemented as a gate array in CMOS (it is manufactured by Mostek), a proven technology which assures both high wafer yields and ready availability for production. As a CMOS device it consumes only 75 mA at 8 MHz and the entire Development Board (including 24K of high speed RAM) requires less than 1 amp.

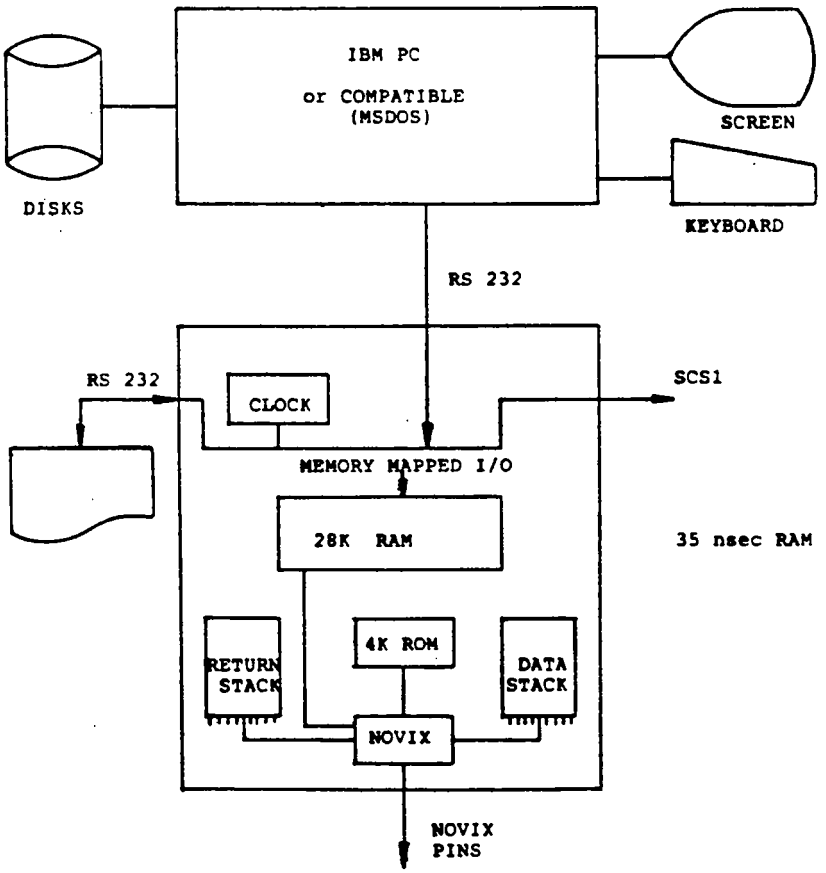
A further consequence of the use of CMOS is that the device is static internally, so that it can be single stepped for hardware debugging purposes if needed.

The Development Environment

The development system for the NC4000 is called the Beta Board. It consists of an independent single board computer that contains:

NOVIX NC4000P
4K words of ROM
28K words of 35 ns RAM
Two memory mapped serial ports
An on board clock for execution timing
Connection for breadboard with access to full Novix Bus

BETA BOARD DEVELOPMENT SYSTEM



In addition, software is provided to run under MSDOS on an IBM PC or look-alike that will use the PC as a terminal and supporting disk drive to the NOVIX.

The software supplied is a FORTH Inc polyFORTH system. It supports multi-tasking (5 microsec task switch time) and can generate standalone ROM based code that does not need to include the development support sections. The compiler is optimised to the NOVIX in that it will combine multiple FORTH words into a single NC4000 instruction where possible.

The Benchmarks and Conclusions

At the time of writing, the Beta Board on which these tests are to be run is still awaiting an export licence. Details of the machine's performance will be distributed at the Conference.

References

1. Programmers Introduction to the NOVIX NC4000P Microprocessor by L Brodie - Published by NOVIX.

2. Fast Processor chip takes its instruction directly from FORTH
   Golden, Moore, Brodie.
   Electronics Design, March 1985.