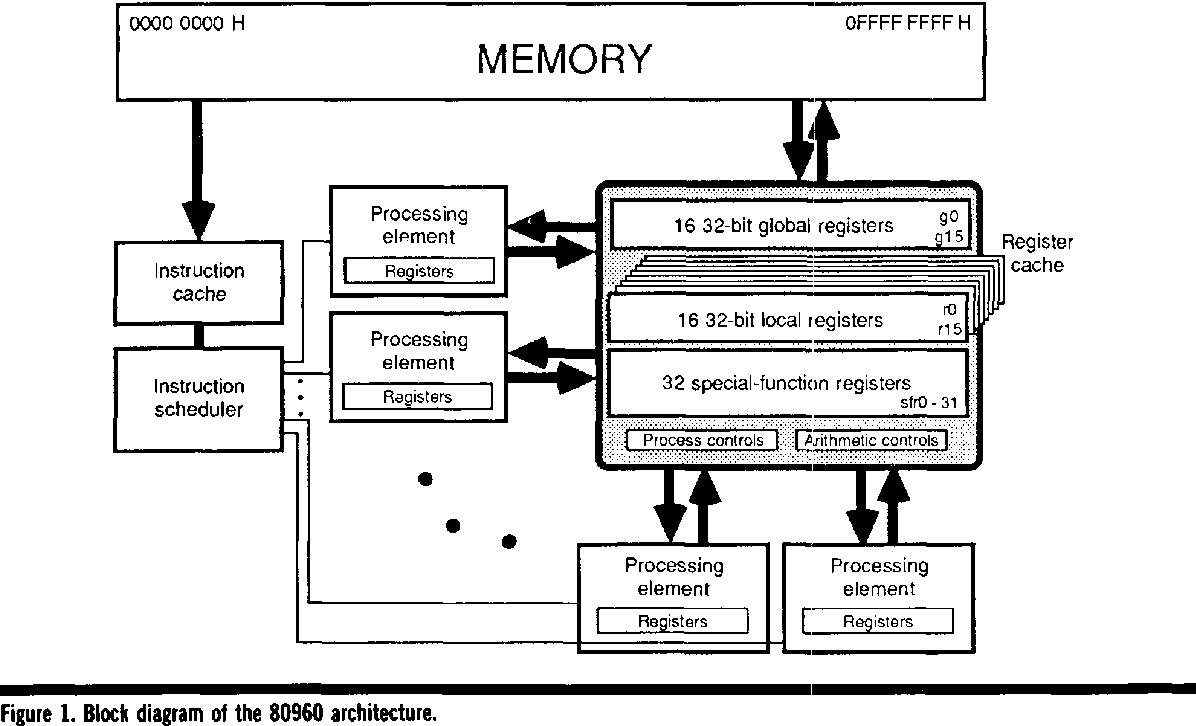
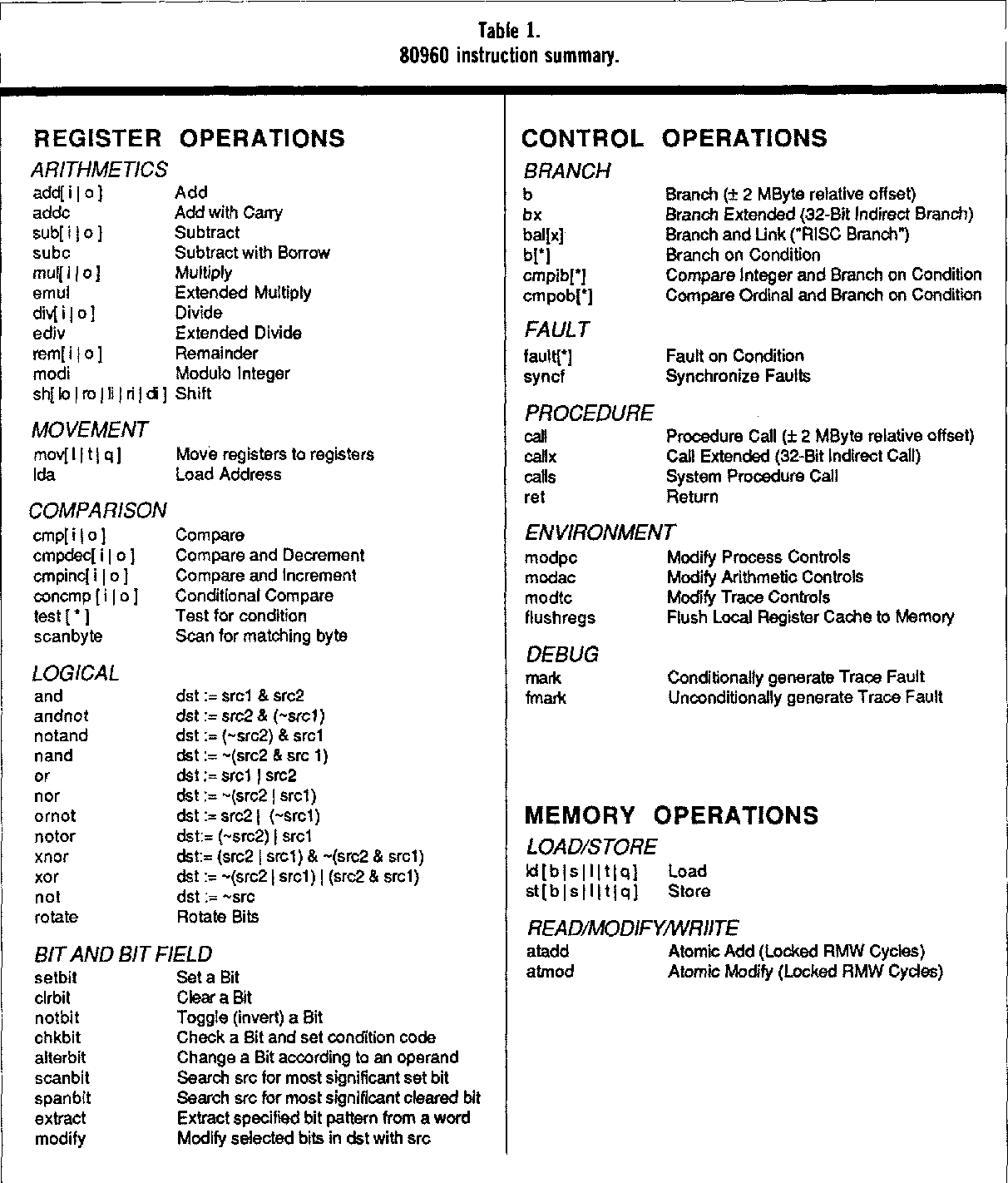
Introduction:

[Intel](https://en.wikipedia.org/wiki/Intel)'s **i960** (or **80960**) was a [RISC](https://en.wikipedia.org/wiki/RISC)-based [microprocessor](https://en.wikipedia.org/wiki/Microprocessor) design that became popular during the early 1990s as an [embedded](https://en.wikipedia.org/wiki/Embedded_system) [microcontroller](https://en.wikipedia.org/wiki/Microcontroller). It became a best-selling CPU in that segment, along with the competing [AMD 29000](https://en.wikipedia.org/wiki/AMD_29000).[[2]](https://en.wikipedia.org/wiki/Intel_i960#cite_note-2) In spite of its success, Intel stopped marketing the i960 in the late 1990s, as a result of a settlement with [DEC](https://en.wikipedia.org/wiki/Digital_Equipment_Corporation) whereby Intel received the rights to produce the [StrongARM](https://en.wikipedia.org/wiki/StrongARM" \o "StrongARM) CPU. The processor continues to be used for a few military applications.

Architecture:





The i960 can access memory as 8, 16, 32,64,96, or 128-bit quantities. For 8, 16, and

32-bit accesses, *integer* load/store instructions will sign extend the most signifIcant bit as

appropriate. *Ordinal* instruction are provided for unsigned data. The larger accesses (64,96, and 128-bit) are useful for manipulation of floating point values, data structures, and provide

an excellent match to the hardware memory burst capability ofthe i960. The i960CA exploits

this data movement by allowing access to an on-chip SRAM through a 128-bit data bus,

allowing up to four registers to be accessed in a single cycle.

A third class of instructions, called CTRL (for control), provide all flow control and

calVreturn. CTRL instructions also may execute independently and concurrently with other instructions in the processor. Unlike other RISC architectures, the i960 does not utilize delay

slots after branches. Because the i960 architecture supports superscalar implementations

(and hence the ability for several instructions to be in the same pipeline slot), the number of

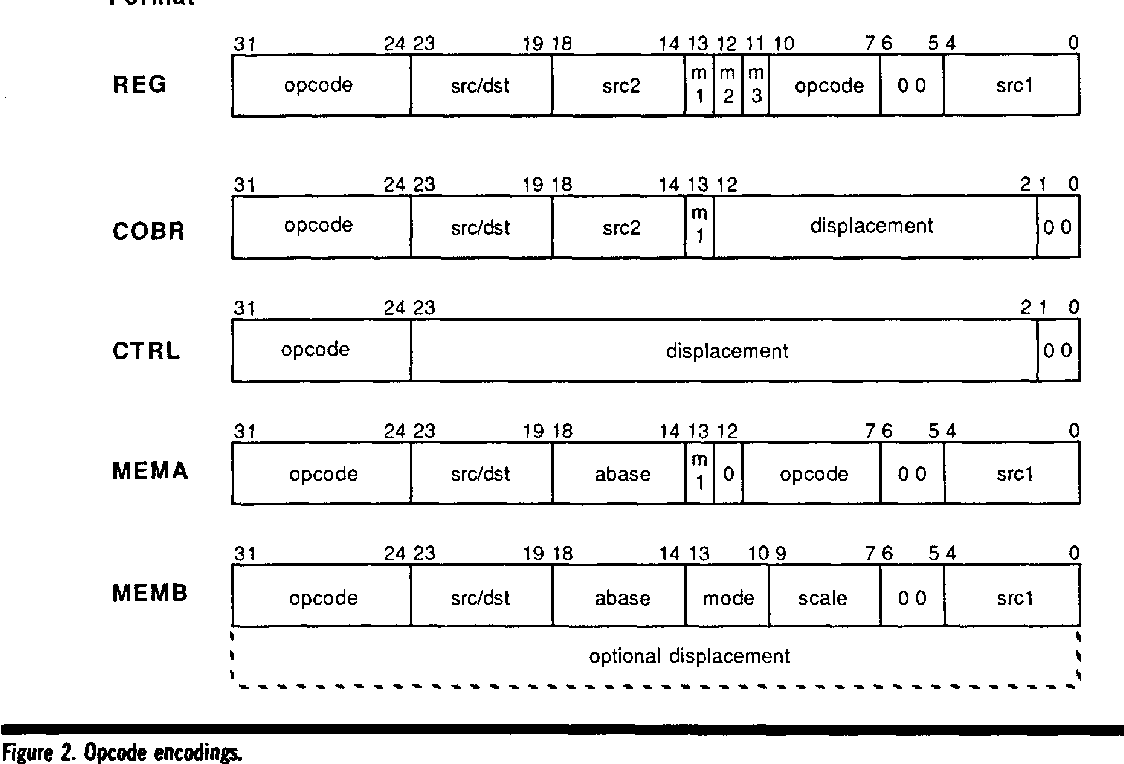
delay slots could not easily be fixed across all parts. The architects chose an alternative

method, called *branch prediction.* Branch prediction takes the form of a bit in the CTRL

opcode which provides the processor a hint of the *usual* path through the branch (i.e. taken

or not taken). The control logic of the processor loads the pipeline from this path, and if the

hint was correct, the processor incurs no penalty. If the hint was wrong, a pipeline stall occurs.



Instruction format.