AccessionIndex: TCD-SCSS-T.20250919.001
Accession Date: 19-Sep-2025
Accession By: Dr.Brian Coghlan
Object name: Motorola MC14500 microprocessor
Vintage: 1977
Synopsis: A very interesting 1-bit microprocessor designed for industrial control applications.

Description:
The Motorola MC14500 [1][2][3] was intended for simple industrial control logic applications. It was particularly well suited for the implementation of ladder logic. The chip only contained about 500 transistors (Fig.1) in a 16-pin package (Fig.2).

It had a 1-bit datapath, but executed any of sixteen 4-bit instructions, see Fig.3, these being presented on a 4-bit opcode path. The program counter and program memory had to be implemented externally, presenting each new instruction to the 4-bit opcode path on the positive transition of the clock. The size of the program address space was therefore at the discretion of the designer.

Architecturally it only had one 1-bit RESULT register RR storing the result from the 1-bit ALU, although internally it had a 4-bit instruction register, and various flag registers to indicate specific status and instruction events, see Fig.4. It has been reported that its architecture was similar to that of the DEC PDP-14 computer.

The flag registers were:

Flag	**Activation**
JMP	logic high during the execution of a JMP instruction
RTN	logic high during the execution of an RTN instruction
FLAG 0	logic high while executing a NOP0 (0x00) instruction
FLAG F	logic high while executing a NOPF (0xff) instruction

Support for JMP had to be implemented externally. For applications that underwent a simple loop, a JMP instruction at the end of the loop could activate the JMP pin to reset the program counter to 0 (the start of the loop) as per Fig.5. Urs Lindegger has developed an MC14500 simulator for the simple system of Fig.5 [4]. Note however that the instruction was semantically neutral, and could be used just to activate the JMP pin for whatever purpose the designer desired. And also note that program branches could be handled completely by external logic independently of the JMP instruction.

Similarly (if used) support for RTN had to be implemented externally, bearing in mind that the RTN instruction was also semantically neutral, and could be used just to activate the RTN pin for whatever purpose the designer desired. Again also note that program subroutine calls and returns could be handled completely by external logic independently of the RTN instruction.

The same semantic neutrality was true for both the NOPO and the NOPF instructions. The designer had complete freedom in utilising the instructions and outputs.

It was a fully static design fabricated in CMOS, executing one instruction per clock cycle. It had a built-in RC clock generator, its frequency being determined using a simple external resistor, although it could also be fed with an external clock. Any clock frequency could be used from DC to 1.0MHz. It operated with a power supply voltage range of 3V to 18V, making it very suitable for industrial programmable logic controllers (PLCs).

For demonstration and learning there is an excellent Arduino shield for Motorola MC14500 available that enables execution of software by the 14500, see [5] and elsewhere in this Collection. For another interesting demonstration board see [6].

Many thanks to Brian Coghlan for donating this item.

The homepage for this catalog is at: https://www.scss.tcd.ie/SCSSTreasuresCatalog/
Click *'Accession Index'* (1st column listed) for related folder, or *'About'* for further guidance.
Some of the items below may be more properly part of other categories of this catalog,
but are listed here for convenience.

Accession Index	Object with Identification
TCD-SCSS-T.20250919.001	Motorola MC14500 microprocessor. A very interesting 1-bit microprocessor designed for industrial control applications. 1977.
TCD-SCSS-T.20250919.001.01	Motorola MC14500BCP microprocessor.
TCD-SCSS-T.20250919.002	Arduino shield for Motorola MC14500. A board that enables execution of software by the 14500, a very interesting 1-bit microprocessor designed for industrial control applications, 2025.
TCD-SCSS-X.20250916.001	Dr.Brian Coghlan's Collection of Early Microprocessors. An extensive and nearly complete set of unused 1970s microprocessor chips, most accompanied with documentation, some with demonstration boards. 1971.

References:

1. Wikipedia, *Motorola MC14500*, see:
   https://en.wikipedia.org/wiki/Motorola_MC14500
   Last browsed to on 19-Sep-2025.

2. Motorola, *MC14500B datasheet*, see:
   https://www.scss.tcd.ie/SCSSTreasuresCatalog/hardware/TCD-SCSS-T.20250919.001/Motorola-MC14500B-datasheet-MC14500Brev3.pdf
   Last browsed to on 19-Sep-2025.

3. Vernon Gregory, Brian Dellande, Ray DiSilvestro, Terry Malarkey, Phil Smith, Mike Hadley, *Motorola MC14500B Industrial Control Unit Handbook*, 33-B78/8.0, Motorola Semiconductor Products Inc., 1977, see:
   https://www.bitsavers.org/components/motorola/14500/MC14500B_Industrial_Control_Unit_Handbook_1977.pdf
   Also: https://www.scss.tcd.ie/SCSSTreasuresCatalog/hardware/TCD-SCSS-T.20250919.001/Motorola-MC14500B-IndustrialControlUnit-Handbook-1977.pdf
   Last browsed to on 19-Sep-2025.

4. Urs Lindegger, *MC14500 Simulator*, see:
   https://www.linurs.org/mc14500.html
   Last browsed to on 19-Sep-2025.

5. Erturk Kocalar, *Retroshield 14500*, see:
   https://www.tindie.com/products/8bitforce/retroshield-MC14500-for-arduino-mega/
   Last browsed to on 19-Sep-2025.

6. Nicola Cimmino, *PLC14500-Nano 1-bit Single Board Computer REV.D*, see:
   https://www.tindie.com/products/nicola_cimmino/plc14500-nano-1-bit-single-board-computer-revd/
   Also: https://github.com/nicolacimmino/PLC-14500
   Last browsed to on 19-Sep-2025.

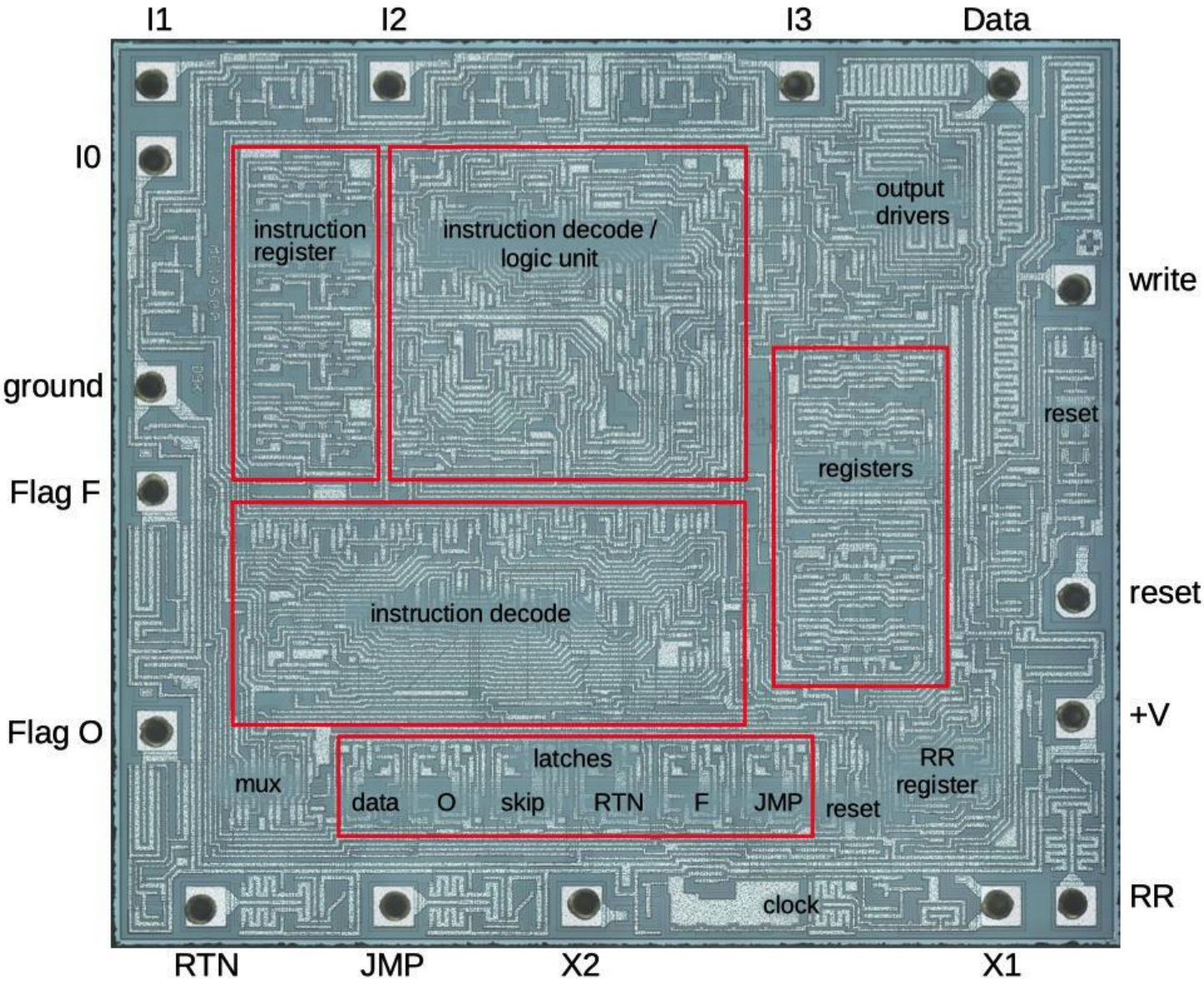

Figure 1: Motorola 14500 chip die micrograph.
(from https://www.righto.com/2021/02/a-one-bit-processor-explained-reverse.html)

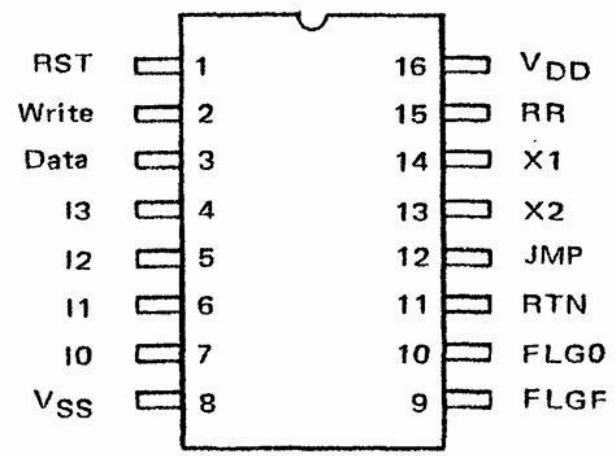

Figure 2: Motorola 14500 pinout (from Motorola).

Instruction Code		Mnemonic	Action
0	0000	NOPO	No change in registers. RR → RR, Flag O → ⎍
1	0001	LD	Load result register. Data → RR
2	0010	LDC	Load complement. $\overline{\text{Data}}$ → RR
3	0011	AND	Logical AND. RR • Data → RR
4	0100	ANDC	Logical AND complement. RR • $\overline{\text{Data}}$ → RR
5	0101	OR	Logical OR. RR + Data → RR
6	0110	ORC	Logical OR complement. RR + $\overline{\text{Data}}$ → RR
7	0111	XNOR	Exclusive NOR. If RR = Data, RR → 1
8	1000	STO	Store. RR → Data Pin, Write → ⎍
9	1001	STOC	Store complement. $\overline{\text{RR}}$ → Data Pin, Write → ⎍
A	1010	IEN	Input enable. Data → IEN Register
B	1011	OEN	Output enable. Data → OEN Register
C	1100	JMP	Jump. JMP Flag → ⎍
D	1101	RTN	Return. RTN Flag → ⎍ and skip next instruction
E	1110	SKZ	Skip next instruction if RR = 0
F	1111	NOPF	No change in registers. RR → RR, Flag F → ⎍

Figure 3: Motorola 14500 rear view (from Motorola).

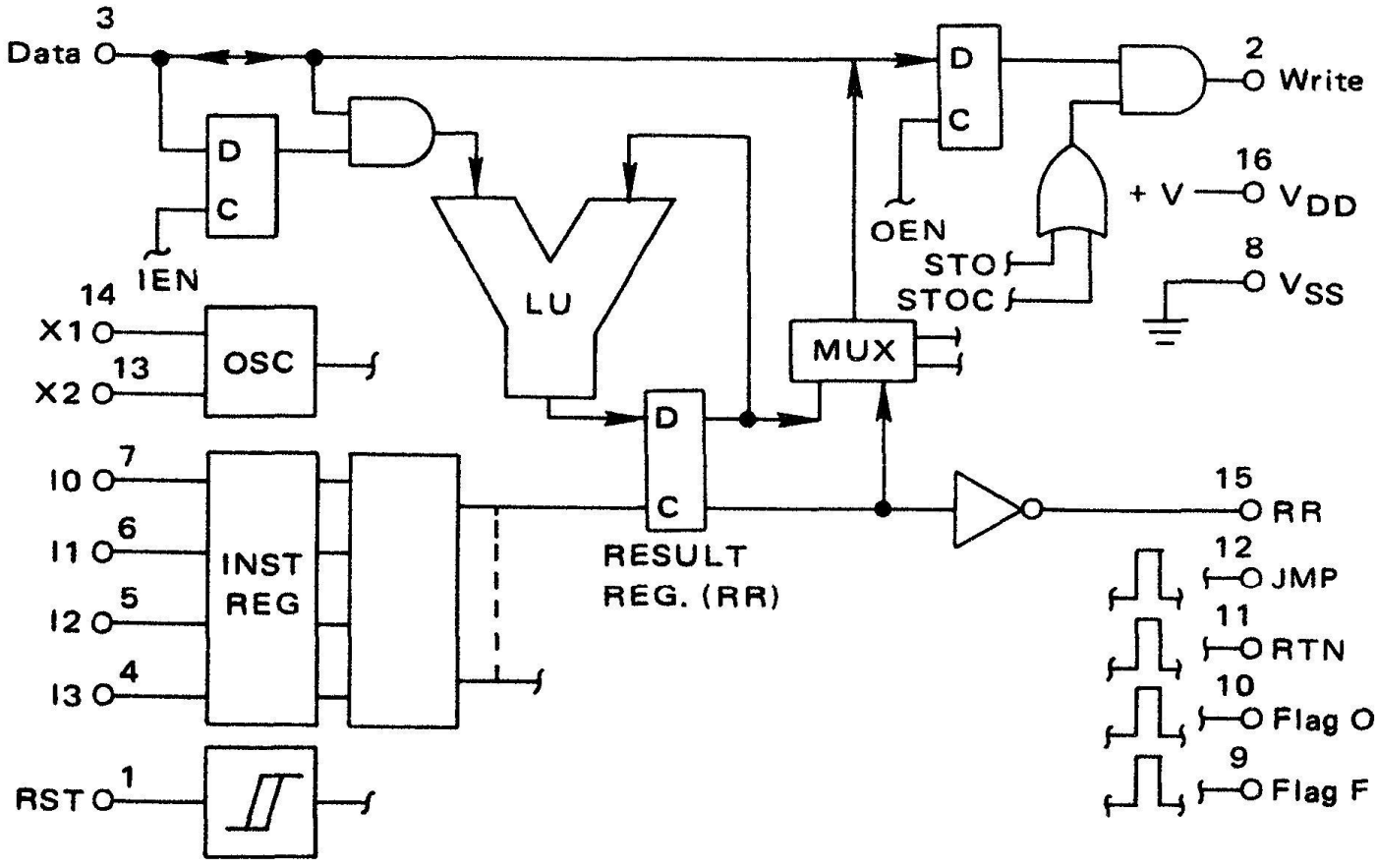

Figure 4: Motorola 14500 architecture (from Motorola).

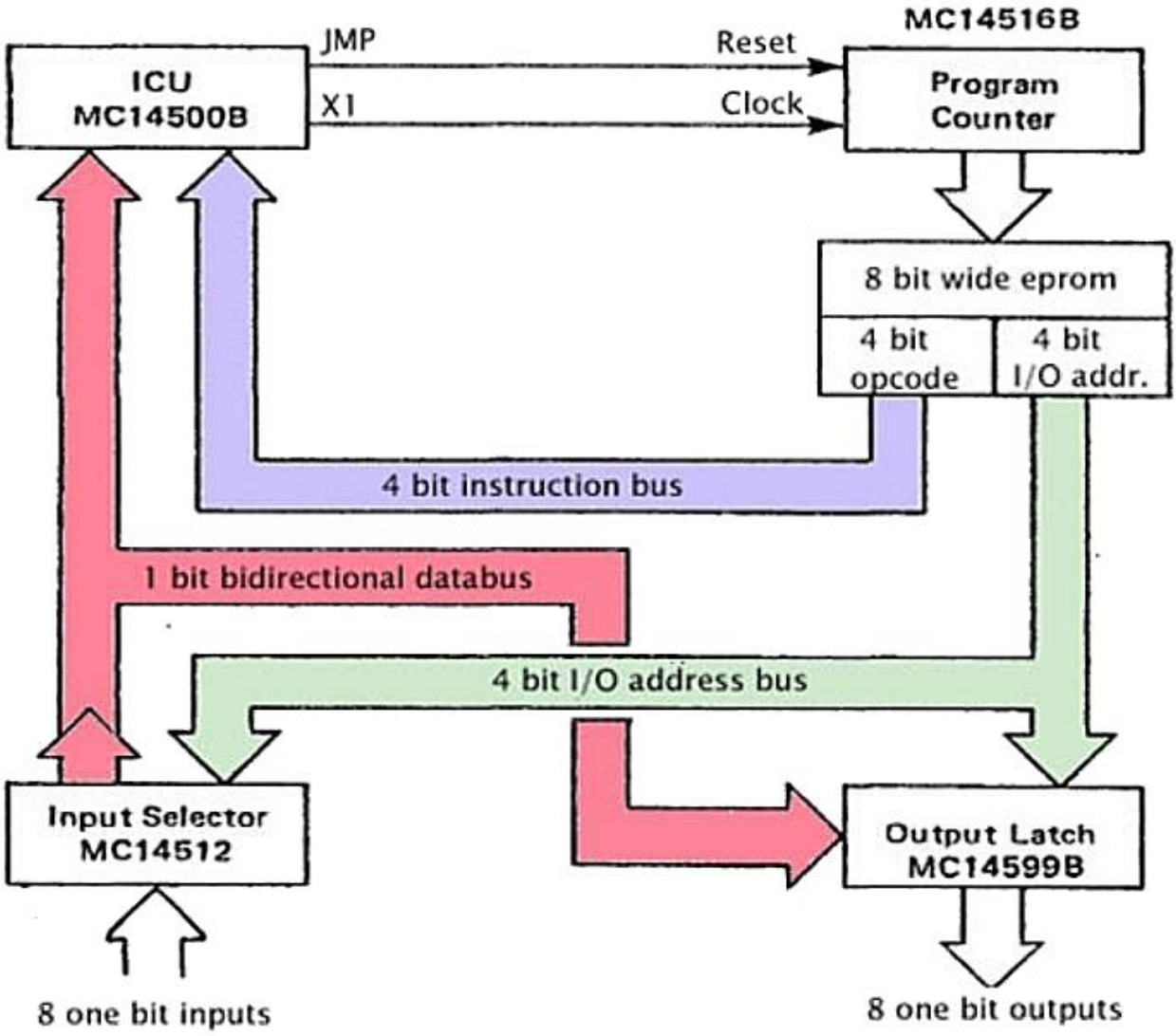

Figure 5: Simple Motorola 14500 system.