AccessionIndex: TCD-SCSS-T.20251114.001
Accession Date: 14-Nov-2025
Accession By: Dr.Brian Coghlan
Object name: National Semiconductor IMP-16 microprocessor and associated chips
Vintage: 1973
Synopsis: A 4-bit bit-slice version of the Data General Nova minicomputer.

Description:
National Semiconductor's *General Purpose Controller / Processor* (GPC/P) was introduced in Q1/1973. It was the first bit-slice microprogrammable microprocessor, partitioned into two chips, a standard 4-bit arithmetic slice and a customized control read-only memory [1] (Figs.1-4). The GPC/P architecture was inspired by that of the Data General Nova minicomputer. It appears that users avoided microprogramming, preferring to use National's own standard microprograms in their IMP-4, IMP-8 and IMP-16 chip sets. These are described below.

A very nice overview is from AntiqueTech [2], although it neglects the other five of the very earliest microprocessors[1] :

The National Semiconductor IMP-16 is actually a two chip implementation, it's heart is being the IMP-00A.The National Semiconductor IMP-00A bit-slice microprocessor was introduced to market in the beginning of 1973, this is just the fourth microprocessor (1st, Intel 4004, 2nd Intel 8008, 3rd Rockwell PPS-4). The IMP-00A is the first bit-slice microprocessor, predating the Intel 3002 and AMD 2901. The IMP-00A was developed for the IMP series of minicomputers. The IMP-16 Minicomputer was implemented using 4 IMP-00A's and a IMP-16A CROM.

CPU devices are comprised on two major components the control unit (CU) and the arithmetic and logic unit (ALU). The CU controls the actions of the CPU and the ALU does the math and logic functions. Bit-slice microprocessors are CPU's and have been split apart into CU and ALU functions. The IMP-00A is the ALU function. The CU can be implemented with custom circuitry, but a sequencing devices such as the IMP-16A is generally used to implement the CU.

National Semiconductor named the IMP-00A a RALU (Register and Arithmetic Logic Unit). The 4-bit IMP-00A was implemented using silicon gate PMOS technology and was clocked at about 700KHz. The IMP-00A had 7 4-bit registers, a status register, and 16 word stack. The IMP-00A could perform ADD, AND, OR, and XOR operations at a rate of about 1 million per second. To support the IMP-00A bit-slice processor, National Semiconductor created a CROM chip. The CROM function was to provide storage for the microprogram and control logic for the IMP-00A (this chip offered in separate auction). Together the CROM and IMP-00A RALU comprise a functional CPU. The IMP-16A CROM was used to implement the IMP minicomputer.

The CROM can hold 100 microinstructions. National Semiconductor created 6 CROM's with pre-defined instruction sets for use with IMP-00A. There were IMP-4A/521 (4-bit standard instruction set), IMP-8A/520 (8-bit standard instruction set), IMP-16A/521 (16-bit standard instruction set), and IMP-16A/522 (16-bit extended instruction set). These pre-defined instructions set were influenced by the Data General Nova minicomputer.

The two CROMs not mentioned were the IMP-16A/523 and IMP-16A/524.

Each IMP-00A/520 *register / arithmetic logic unit* (RALU) [3] had its own 4-bit slice of the microprogram counter, the ALU, status flags (overflow, link, carry, flag), a 16-word LIFO stack, and four accumulators (two able to be used for indexing), memory address and data registers, and an input multiplexer to select ALU status and other status. The *control read-only memory* (CROM) stored 100 x 23-bit microinstructions, as well as program sequencing, subroutine execution, translation of microinstructions into RALU commands, and control logic for up to eight RALUs.

[1] Four Phase Systems AL1 [Apr-1969], Garret AiResearch MP944 [1970], Fairchild PPS-25 [May-1971], Texas Instruments TMX1795 [Jun/Jul-1971], Pico / General Instrument PICO1 / GI250 [1971].

The ALU performed ADD, AND, OR and exclusive OR operations on true and complemented data from the registers. A shifter was provided for single bit left or right shifts and an I/O data multiplexer for communication with a 4-bit external data bus. Control was provided over a 4-bit, time-multiplexed command bus.

The various actions were sequenced at periods T1-8 defined by an ~700kHz 4-phase non-overlapping clock, see Figs.5-9. The chipset operated on +5V and -12V supplies, with two different logic levels. The clocks and signals that interfaced between the RALU and CROM were at active-low pMOS levels (logic 0 = +5V, logic 1 = -12V), while those for interface with the external system were at active-high TTL levels (logic 0 = 0V, logic 1 = +5V). The chips were packaged in ceramic (suffix J) and plastic (suffix D) 24pin DIP.

Internally there were four RALU busses: A, B, R and S. Any of the registers could output on the A or B bus. Then A (or A complemented) and B could be fed to the ALU. Additions could input carry from CSH0 and generate carry to CSH3 at T5. The ALU output on the S bus was fed to the shifter for 1-bit left or right shift (or no shift). The most or least significant shifted bits were conveyed via pins CSH0 or CSH3 at T7-8. The shifter output fed the R bus, from where registers could be updated. The multiplexer could feed the A bus from the external data bus, or feed the A or R bus to the external system.

Commands were 16-bits, multiplexed from the CROM onto the RALU's NCB0-3 inputs during the T1, T3, T5, T7 periods in each RALU cycle (NCB0-3 had to be logic 0 at T2, T4, T6, T8). In T1 the three least-significant bits specified zero or the registers R1-7 to be output to the A bus. If in T1 NCB3 was logic 1 and NCB0-2 was zero the stack was popped onto the A bus. In T3 the three least-significant bits specified zero or the registers R1-7 to be output to the B bus. During T3 and T4 the A bus was fed to the external system, typically used for address and data output to system memory or peripherals. In T5, NCB0-1 specified ALU operations (00:AND, 01:XOR, 10:OR, 11:ADD), while NCB3-2 specified control functions (00:NOOP, 01:R control, 10:shift left, 11:shift right). In T7 the three least-significant bits specified zero or the registers R1-7 to be updated from the R bus. If in T1 NCB3 was logic 1 and in T7 NCB0-2 was zero the R bus was pushed onto the stack. If in T7 NCB3 was logic 1 the R bus was fed from the multiplexer, typically from system memory or peripheral devices. During the subsequent T1 and T2 the R bus (i.e. the result of the last operation) was fed to the external system.

Thus commands were unusually fine-grained. Although it was not explicitly stated in IMP-16 datasheets, in its pure GPC/P form the bit-slice chipset was intended as an underlying hardware (micro-machine) to be microprogrammed with a higher-level instruction set. The IMP-16A/521 CROM [4] provided a standard 16-bit instruction set and programming model (which only included the user-level program counter, status, stack and accumulators) for IMP-16 computers [5][6], see Figs.8-9. Of the RALU's registers, three were reserved for the 16-bit user-level program counter, 16-bit memory address (MAR), and 16-bit memory data (MDR). The other four registers were used as 4-bit user-level accumulators, two of which could be used as index registers (the other two had special implications for certain instructions), and the stack was used for 16-bit user-level return addresses. The microinstruction cycle was 1.4µs,

several being needed to execute typical instructions. A register-to-register addition of two 16-bit numbers took 12µs, while two 16-bit numbers could be multiplied into a 32-bit result in 150µs. This standard instruction set did not support bytes, only 64k x 16-bit words. The optional IMP-16A/522 CROM II (also [4]) implemented seventeen extra instructions, including 16-bit multiply and divide, double-precision add and subtract, set/reset/test bit, and byte operations on the lower 32kB of memory. The IMP-16A/523 CROM [7] provided additional block transfer, context save & restore, stack save & restore, search, and bit-scan instructions. The purpose of the IMP-16A/524 CROM is not known.

The IMP-8A/520 CROM [8] provided an 8-bit instruction set. It assumed the address space was extended to 16-bits with an 8-bit external *page-counter* to hold the most-significant memory address bits. Of the RALU's registers, three were reserved: one for 8-bit memory data, and two others for the least-significant 8-bits of both the memory address and the user-level program counter (thus relative addressing could be used within the current page, but page changes had to be done indirectly via page zero). The other four registers were used as 4-bit user-level accumulators, two of which could be used as index registers, and the stack was used for user-level 8-bit relative return addresses.

The IMP-4A/521 CROM [9] and IMP-7A-540D FILU [10] were used in a specialized 4-bit system with a microprogramming tool called *Field-Alterable Control Element* (FACE) [11] to help designers modify the instruction set. The FILU (*Four-bit Interface Logic Unit*) was the interface to memory and peripherals. It added the logic needed to make the 4-bit CPU work with a 12-bit address bus, with a 12-bit memory address register, 12-bit user-level program counter, 7 x 12-bit return address stack, and an instruction register to address the CROM. Of the RALU's registers, three were reserved for 4-bit constants, the other four were used as 4-bit user-level accumulators, two of which could be used as index registers, while the stack could used for 4-bit user-level data. FACE was implemented on a board, and was connected by cable to the CROM socket of an IMP-16 board to form a writable control store.

It is not known whether any other CROMs were ever defined, despite the architecture supporting up to eight RALUs that could implement a 32-bit machine, however it appears that above 24 bits the performance was affected by propagation delays.

The IMP-16 chipset was used in several computers, mostly using the IMP-16C CPU board [12] (Figs.12-14), which consisted of the microprocessor, clock system, I/O bus drivers, 256 words of RAM and provision for 512 words of ROM/PROM. The standard 16-bit IMP-16 instruction set was implemented using 4 x IMP-00A/520 RALU plus one IMP-16A/521 CROM, with provision to add an IMP-16A/522 CROM II to provide the extended instruction set. One additional feature that was absent from the chipset was vectored interrupts (latency 4.5µs) and non-vectored interrupts (latency 35µs).

The IMP-16P computer [13] (the ‘starter kit’) added a chassis, power supplies, at least one 4k x 16-bit memory module, and teletype, tape, card-reader, printer and PROM programmer interfaces for application software development.

The IMP-16L computer [14] (Figs.15-17) chassis added a front-panel display for access to memory and registers, and usually more memory (up to a limit of 64k words). An asynchronous bus allowed fast direct memory access (DMA) by peripheral devices without the need to transit the RALUs.

The Microsystem 6600 computer [15] seems to have been not more than a ruggedised IMP-16P, but details are very scarce.

The IMP-16C was also used in the Jacquard Systems J100 small business computer [16], the Aston Martin Lagonda (it seems because National Semiconductor's chairman Peter Sprague was a major shareholder in Aston Martin [17]), and in the Seabourne Integrated Diagnostic System (SIDS) that was installed on several river and ocean going vessels.

The IMP-16 was later superseded by the PACE series of single-chip 16-bit microprocessors (see elsewhere in this catalog) [18][19][20][21][22], which had a similar architecture but were not binary compatible.

Many thanks to Brian Coghlan for donating these items.

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.20251114.001	National Semiconductor IMP-16 microprocessor and associated chips. A 4-bit bit-slice version of the Data General Nova minicomputer. 1973.
TCD-SCSS-T.20251114.001.01	6 x NatSemi IMP-00A/520J RALU (4-bit bit-slice register-ALU).
TCD-SCSS-T.20251114.001.02	2 x NatSemi IMP-16A/521J CROM (control ROM).
TCD-SCSS-T.20251114.001.03	2 x NatSemi IMP-16A/522J CROM-II (extended control ROM).
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, *IMP-16*, see:
https://en.wikipedia.org/wiki/IMP-16
Last browsed to on 6-Oct-2025.

2. AntiqueTech, *IMP-16*, see:
http://www.antiquetech.com/?page_id=696
Last browsed to on 6-Oct-2025.

3. National Semiconductor, *IMP-00A/520 MOS/LSI register and arithmetic logic unit (RALU)*, 1977, see:
https://www.scss.tcd.ie/SCSSTreasuresCatalog/hardware/TCD-SCSS-T.20251114.001/NatSemi-IMP16-RALU-IMP-00A-520J-dataSheet-1977-OCR.pdf
Last browsed to on 6-Oct-2025.

4. National Semiconductor, *IMP-16A/521D, IMP-16A/522D, IMP-8A/5200 MOS/LSI control and read only memory unit (CROM)*, see:
https://www.scss.tcd.ie/SCSSTreasuresCatalog/hardware/TCD-SCSS-T.20251114.001/NatSemi-IMP16-CROM-IMP-16A-521D-522D-dataSheet-OCR.pdf
Last browsed to on 6-Oct-2025.

5. National Semiconductor, *IMP-16 Programming Manual*, November 1973, see:
https://www.scss.tcd.ie/SCSSTreasuresCatalog/hardware/TCD-SCSS-T.20251114.001/NatSemi-IMP16-4200002B-IMP16-ProgrammingManual-Nov1973.pdf
Last browsed to on 6-Oct-2025.

6. National Semiconductor, *Use of RALU Flags*, Application Note, June 1972, see:
https://www.scss.tcd.ie/SCSSTreasuresCatalog/hardware/TCD-SCSS-T.20251114.001/NatSemi-IMP16-4200011A-Use-of-RALU-Flags-AppNote-Jun1972.pdf
Last browsed to on 6-Oct-2025.

7. National Semiconductor, *IMP-16A/523 POWRIO datasheet*, 1975, see:
https://www.scss.tcd.ie/SCSSTreasuresCatalog/hardware/TCD-SCSS-T.20251114.001/NatSemi-IMP16-POWRIO-IMP-16A-523-datasheet-1975-OCR.pdf
Last browsed to on 6-Oct-2025.

8. National Semiconductor, *IMP-8A/520D CROM dataSheet*, see:
https://www.scss.tcd.ie/SCSSTreasuresCatalog/hardware/TCD-SCSS-T.20251114.001/NatSemi-IMP8-CROM-IMP-8A-520D-dataSheet-OCR.pdf
Last browsed to on 6-Oct-2025.

9. National Semiconductor, *IMP-4A/521D CROM datasheet*, September 1974, see:
https://www.scss.tcd.ie/SCSSTreasuresCatalog/hardware/TCD-SCSS-T.20251114.001/NatSemi-IMP4-CROM-IMP-4A-521D-datasheet-Sep1974-OCR.pdf
Last browsed to on 6-Oct-2025.

10. National Semiconductor, *IMP-7A/540D FILU datasheet*, September 197, see:
https://www.scss.tcd.ie/SCSSTreasuresCatalog/hardware/TCD-SCSS-T.20251114.001/NatSemi-IMP4-FILU-IMP-7A-540D-datasheet-Sep1974.pdf
Last browsed to on 6-Oct-2025.

11. National Semiconductor, *AN123 Microprogram Development System*, September 1974, see:
https://www.scss.tcd.ie/SCSSTreasuresCatalog/hardware/TCD-SCSS-T.20251114.001/NatSemi-IMP16-AN123-MicroprogramDevelopmentSystem-Sep1974.pdf
Also: https://www.scss.tcd.ie/SCSSTreasuresCatalog/hardware/TCD-SCSS-T.20251114.001/NatSemi-computer-uSpec-6-MicroprogramDevelopmentSystem-1975.pdf
Last browsed to on 6-Oct-2025.

12. National Semiconductor, *IMP-16C*, see:
https://www.scss.tcd.ie/SCSSTreasuresCatalog/hardware/TCD-SCSS-T.20251114.001/NatSemi-computer-IMP16C-InstructionGuide.pdf
Last browsed to on 6-Oct-2025.

13. National Semiconductor, *IMP-16P*, see:
https://www.scss.tcd.ie/SCSSTreasuresCatalog/hardware/TCD-SCSS-T.20251114.001/NatSemi-computer-IMP16P-4200036A-IMP16P-Description-1974.pdf
Last browsed to on 6-Oct-2025.

14. National Semiconductor, *IMP-16L*, see:
https://www.scss.tcd.ie/SCSSTreasuresCatalog/hardware/TCD-SCSS-T.20251114.001/NatSemi-computer-IMP16L-4200028X-IMP16L-UsersManualPreliminary-Apr74.pdf
Last browsed to on 6-Oct-2025.

15. National Semiconductor, *Microsystem 6600*, see:
https://www.scss.tcd.ie/SCSSTreasuresCatalog/hardware/TCD-SCSS-T.20251114.001/NatSemi-computer-Microsystem6600-SchoolCompSci-CarletonUniv-Ottawa.pdf
Last browsed to on 6-Oct-2025.

16. Esther Surden, *Jacquard Systems starts small but thinks big*, p.66, Computerworld, 1977, see:
https://www.scss.tcd.ie/SCSSTreasuresCatalog/hardware/TCD-SCSS-T.20251114.001/JacquardSystems-J100-Computerworld-p66-21Nov1977.pdf
Last browsed to on 6-Oct-2025.

17. Peter Sprague, *Aston Martin*, see:
https://www.scss.tcd.ie/SCSSTreasuresCatalog/hardware/TCD-SCSS-T.20251114.001/NatSemi-IMP16-and-AstonMartin-PeterSprague.pdf
Last browsed to on 6-Oct-2025.

18. Wikipedia, *National Semiconductor PACE*, see:
https://en.wikipedia.org/wiki/National_Semiconductor_PACE
Last browsed to on 6-Oct-2025.

19. National Semiconductor, *PACE-1 IPC-16A/520D datasheet*, 1974, see:

https://www.scss.tcd.ie/SCSSTreasuresCatalog/hardware/TCD-SCSS-T.20251006.004/NatSemi-IMP-00A-520-RALU-datasheet.pdf
Also: https://www.scss.tcd.ie/SCSSTreasuresCatalog/hardware/TCD-SCSS-T.20251006.004/NatSemi-IMP16-databook-4200036A-1974.pdf
Also: https://www.scss.tcd.ie/SCSSTreasuresCatalog/hardware/TCD-SCSS-T.20251006.004/NatSemi-IMP-16C-board.pdf
Last browsed to on 6-Oct-2025.

20. National Semiconductor, *PACE-1 IPC-16A/520D datasheet*, 1974, see: https://www.scss.tcd.ie/SCSSTreasuresCatalog/hardware/TCD-SCSS-T.20251006.004/NatSemi-PACE1-IPC-16A-520D-dataSheet-1974.pdf
Last browsed to on 6-Oct-2025.

21. National Semiconductor, *PACE-1 Assembly Language*, 1974, see: https://www.scss.tcd.ie/SCSSTreasuresCatalog/hardware/TCD-SCSS-T.20251006.004/NatSemi-PACE-AssemblyLanguage-ProgrammingManual.pdf
Last browsed to on 6-Oct-2025.

22. National Semiconductor, *PACE-2 INS8900 datasheet*, 1977, see: https://www.scss.tcd.ie/SCSSTreasuresCatalog/hardware/TCD-SCSS-T.20251006.004/NatSemi-PACE2-INS8900-dataSheet-1977.pdf
Last browsed to on 6-Oct-2025.

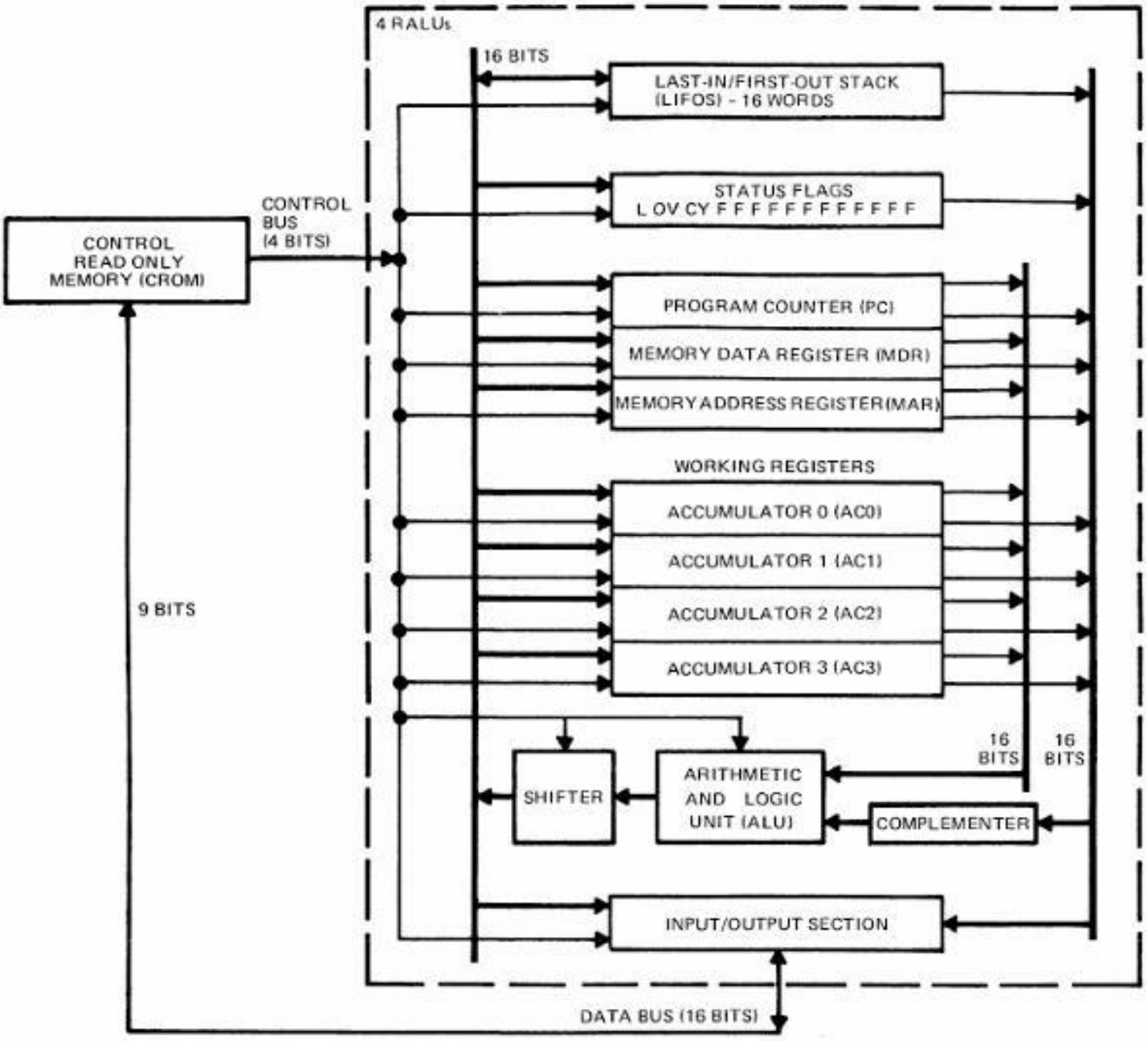


Figure 1: NatSemi GPC/P and IMP-16 microprocessor with 4 x RALU and one CROM.

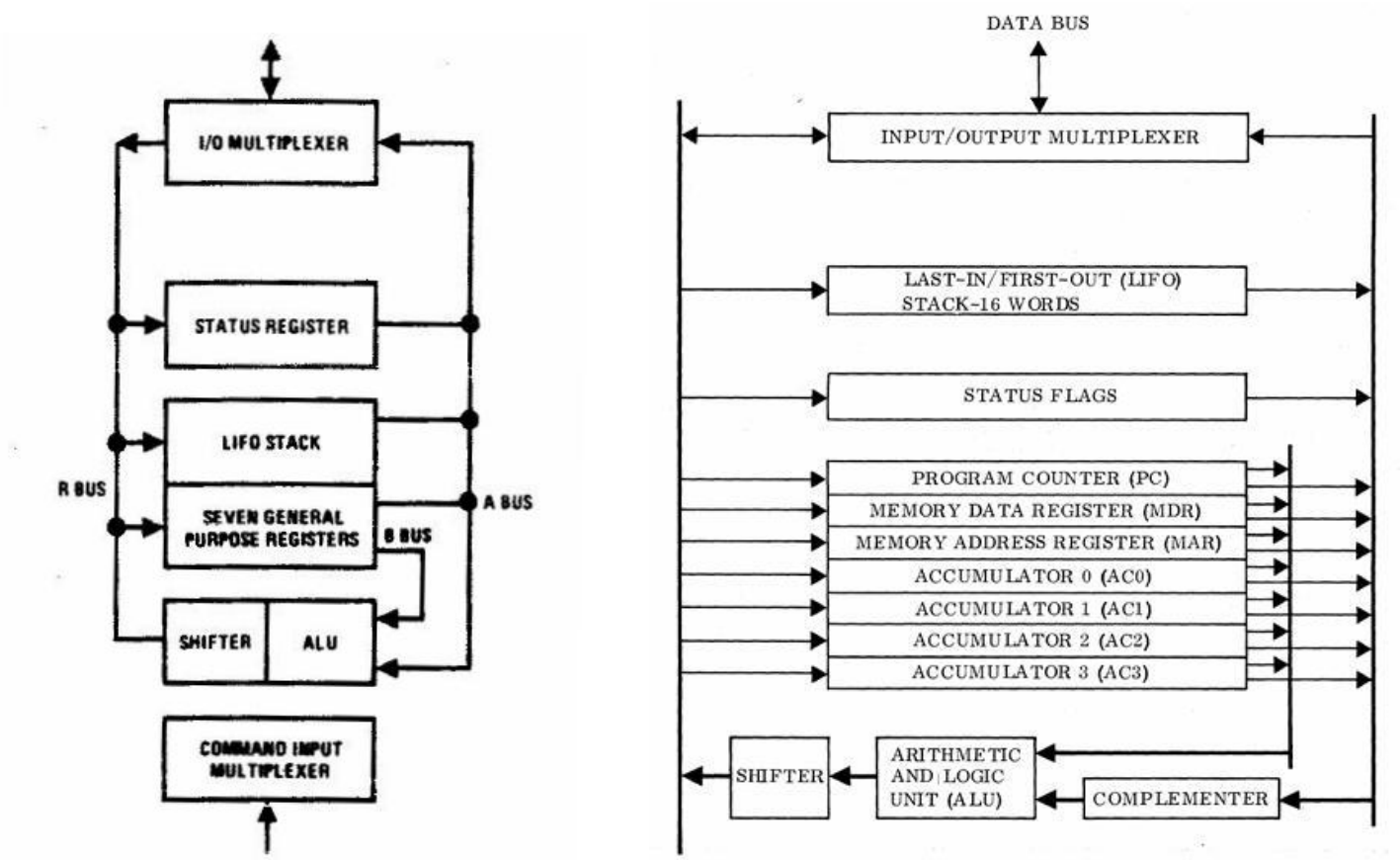


Figure 2: Simplified NatSemi GPC/P and IMP-16 register / arithmetic logic unit (RALU) architecture.

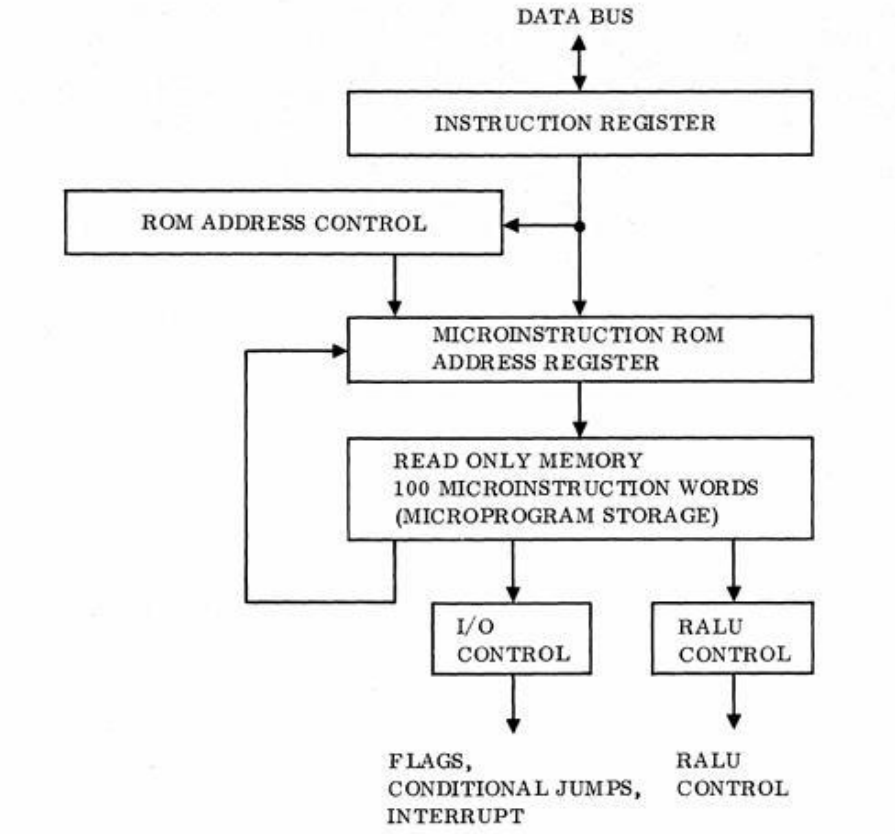


Figure 3: NatSemi GPC/P and IMP-16 control ROM (CROM) architecture.

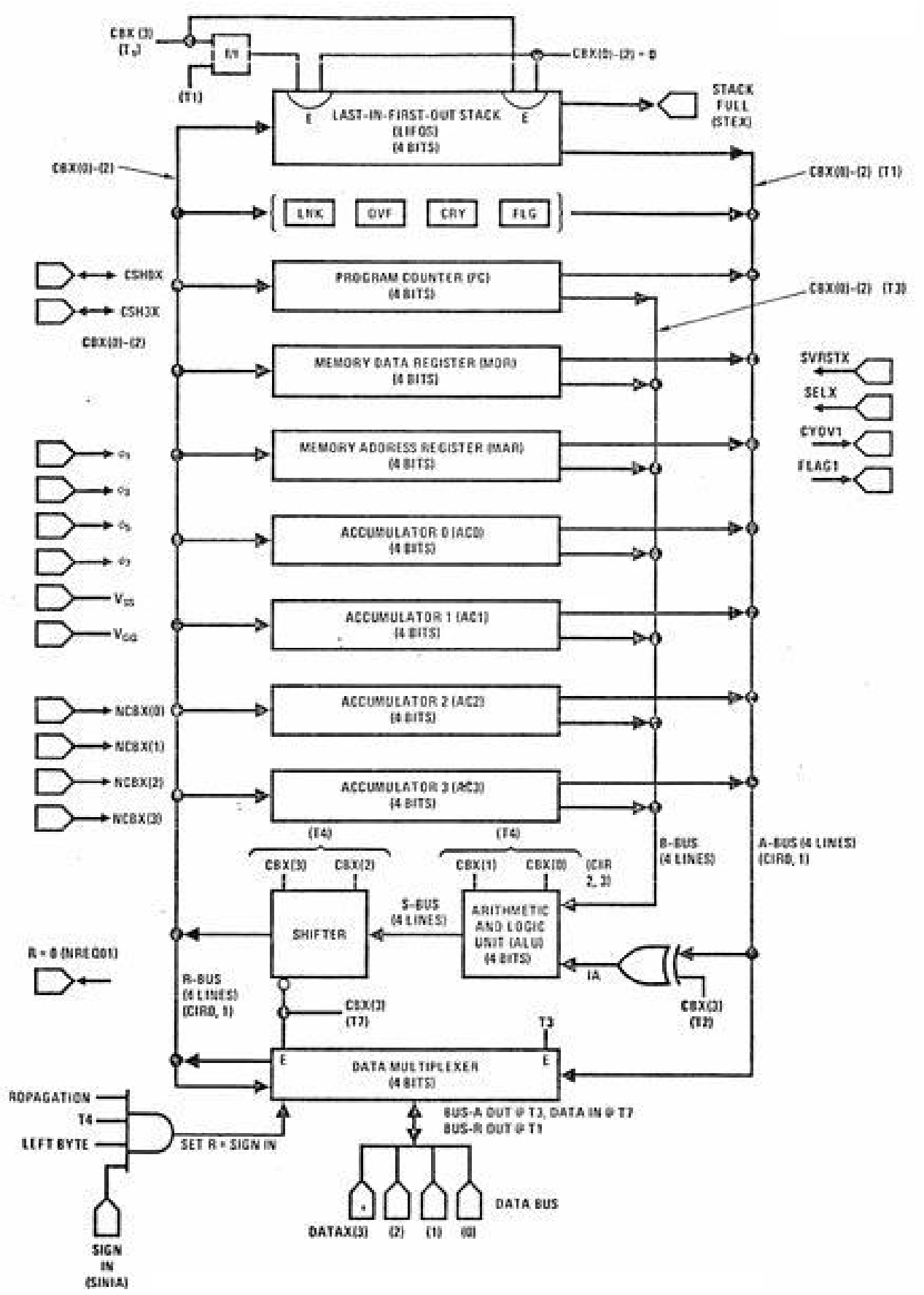
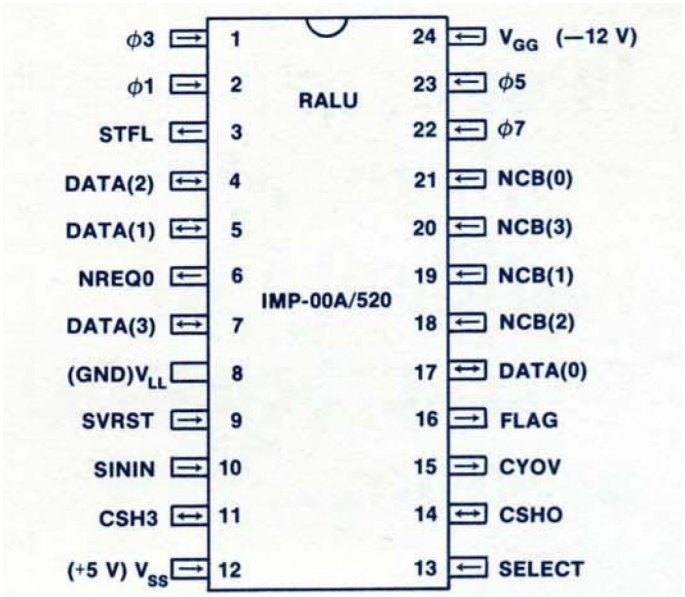
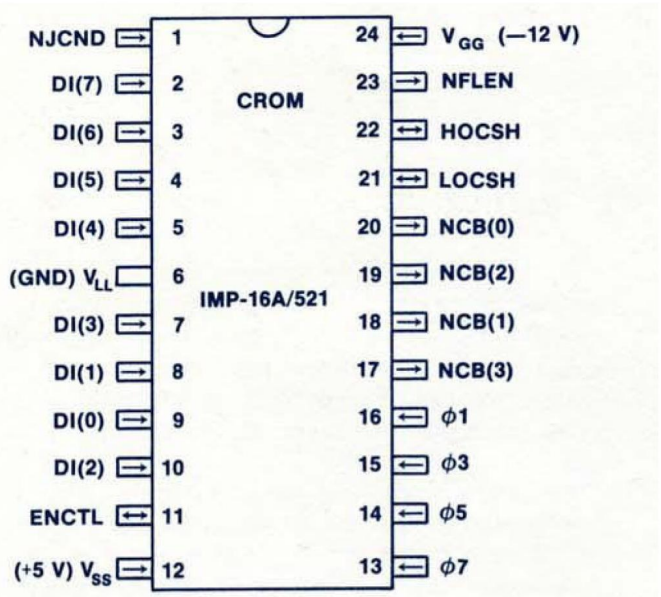


Figure 4: National Semiconductor GPC/P and IMP-16 RALU internal architecture in detail





PIN NO.	FUNCTION
1, 2, 22, 23	4-phase, non-overlapping clock signals.
3	Stack full output goes true when bottom word of stack is non-zero at start of previous cycle.
4, 5, 7, 17	Transfers data between RALU and memory or peripheral devices.
6	Result bus equals zero. Logic level goes to "0" level when R-bus contains all zeroes.
9	Save/Restore line. Provides a means of modifying status flags over the data bus.
10	SININ accomplishes sign extension.
11, 14	Carry input (CSH0) and carry output (CSH3) used primarily for transfer of carry and shift information between RALU's or between RALU and CROM.
13	Select flag input used to select carry or overflow and to determine whether the Link is included in shift operations.
15	Carry/Overflow provides output signal indicating state of Carry flag or Overflow flag as selected by the select input.
16	Flag output indicates state of the general purpose status flag.
18-21	Command inputs to RALU.

PIN NO.	FUNCTION
1	Jump condition input from external jump condition multiplexer.
2-5 & 7-10	Data input lines carry instructions to CROM instruction register.
11	Enable control provides logic "1" when branch to instruction fetch rountine occurs. Responds to logic "1" input by executing branch to instruction fetch routine.
13-16	4-phase, non-overlapping clock signals.
17-20	Command bits to RALU.
21	Low-order carry/shift to and from RALU.
22	High-order carry/shift to and from RALU.
23	Flag enable control output used to set/reset flags addressed by data inputs.

Figure 5: NatSemi IMP-16 RALU and CROM pinouts (from Hewlett-Packard).

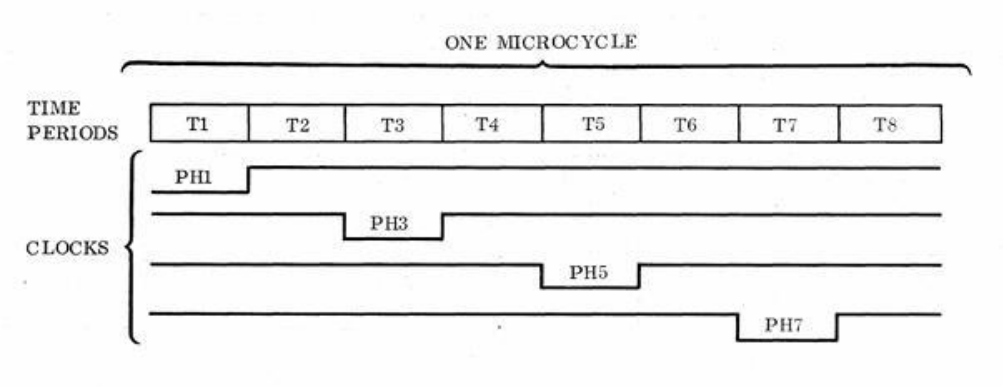


Figure 6: NatSemi IMP-16 RALU four-phase non-overlapping clock cycle.

SIGNALS[1]		LOGIC LEVELS	TIME INTERVALS T1	T2	T3	T4	T5	T6	T7	T8	PIN FUNCTION	PIN NO.
CLOCKS	ϕ_1	MOS									IN	2
	ϕ_3	MOS									IN	1
	ϕ_5	MOS									IN	23
	ϕ_7	MOS									IN	22
COMMAND												
	NCB(0)	MOS	$\overline{A0}$	"0"	$\overline{B0}$	"0"	$\overline{ALU0}$	"0"	$\overline{R0}$	"0"	IN	21
	NCB(1)	MOS	$\overline{A1}$	"0"	$\overline{B1}$	"0"	$\overline{ALU1}$	"0"	$\overline{R1}$	"0"	IN	19
	NCB(2)	MOS	$\overline{A2}$	"0"	$\overline{B2}$	"0"	$\overline{CTL0}$	"0"	$\overline{R2}$	"0"	IN	18
	NCB(3)	MOS	$\overline{STACK}$	"0"	$\overline{COMP}$	"0"	$\overline{CTL1}$	"0"	$\overline{I/O}$	"0"	IN	20
DATA												
	DATA(0),(1),(2),(3)	TTL	R BUS(OUT)			A BUS(OUT)			DATA INPUT	"1" (OUT)	I/O	17,5,4,7
CONTROL												
	SELECT	TTL	Don't Care (DC)				SELECT		DC		IN	13
	SVRST	TTL	DC				SVRST		DC		IN	9
	CYOV	TTL	CARRY OR OVERFLOW						"1"		OUT	15
	FLAG	TTL	FLAG				"1"				OUT	16
	STFL	TTL	STACK FULL					"0"		"1"	OUT	3
	NREQ00	TTL	$\overline{R=0}$				"1"		"0"	$\overline{R=0}$	OUT	6
MISC												
	SININ	MOS-T5 TTL-T7	Don't Care (DC)				BYTE	DC	SIGN	DC	IN	10
	CSH0	MOS	"1" (OUT)	HIGH IMPEDANCE[2]			CARRY (IN)	"1" (OUT)	$\overline{SHIFT}$ I/O		I/O	14
	CSH3	MOS	$\overline{OVCEN}$ (IN)	HIGH IMPEDANCE 2		"0" (OUT)	CARRY (OUT)	"1" (OUT)	$\overline{SHIFT}$ I/O		I/O	11

Note 1: A positive true logic convention is used for all signals (i.e., "1" = more positive voltage, "0" = more negative voltage). Signal names beginning with N are complemented signals.

Note 2: CSH0 and CSH3 high impedance state for intervals T_2 through T_4 is the TRI-STATE mode for output drivers.

Note 3: "1" (OUT) means RALU is driving this node to the "1" logic level during the defined interval. For bidirectional I/O lines the logic state is defined as "in" or "out."

Figure 7: NatSemi IMP-16 RALU instruction cycle.

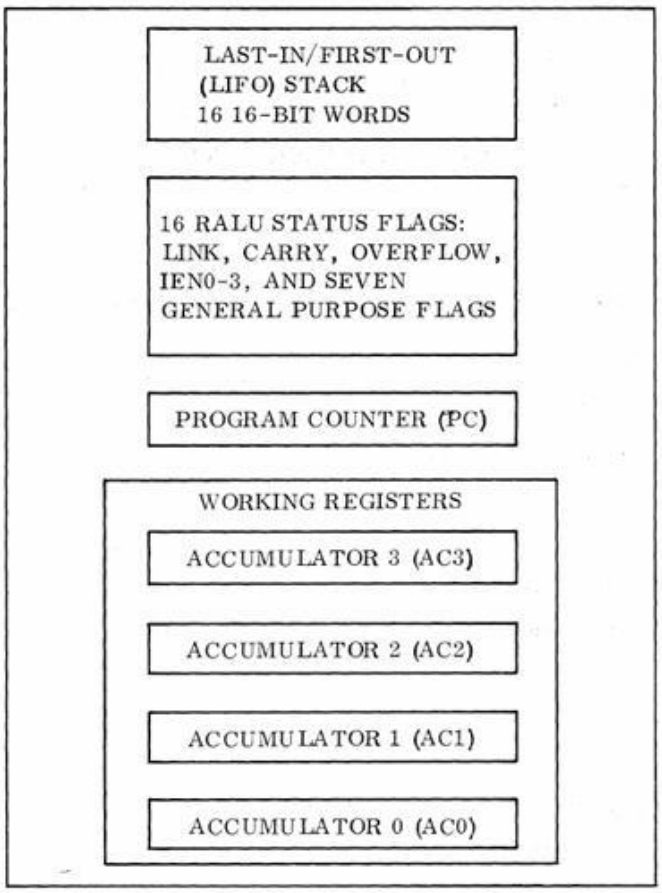


Figure 8: NatSemi IMP-16 user-level registers (from National Semiconductor).

$^{1}_{5}$ $^{1}_{4}$ $^{1}_{3}$ $^{1}_{2}$ $^{1}_{1}$ $^{1}_{0}$ $^{0}_{9}$ $^{0}_{8}$ $^{0}_{7}$ $^{0}_{6}$ $^{0}_{5}$ $^{0}_{4}$ $^{0}_{3}$ $^{0}_{2}$ $^{0}_{1}$ $^{0}_{0}$ *(bit position)*

Main registers

AC0	Accumulator
AC1	Accumulator
AC2	Acc/Base
AC3	Acc/Base

Program counter

PC	Program Counter

Stack

STK	(16 entries)

Status Flags Register (FR)

L	OV	CY	GF	GF	GF	GF	GF	GF	GF	GF	GF	GF	GF	GF	GF	Status

Figure 9: NatSemi IMP-16 microprocessor user-level registers (from Wikipedia).

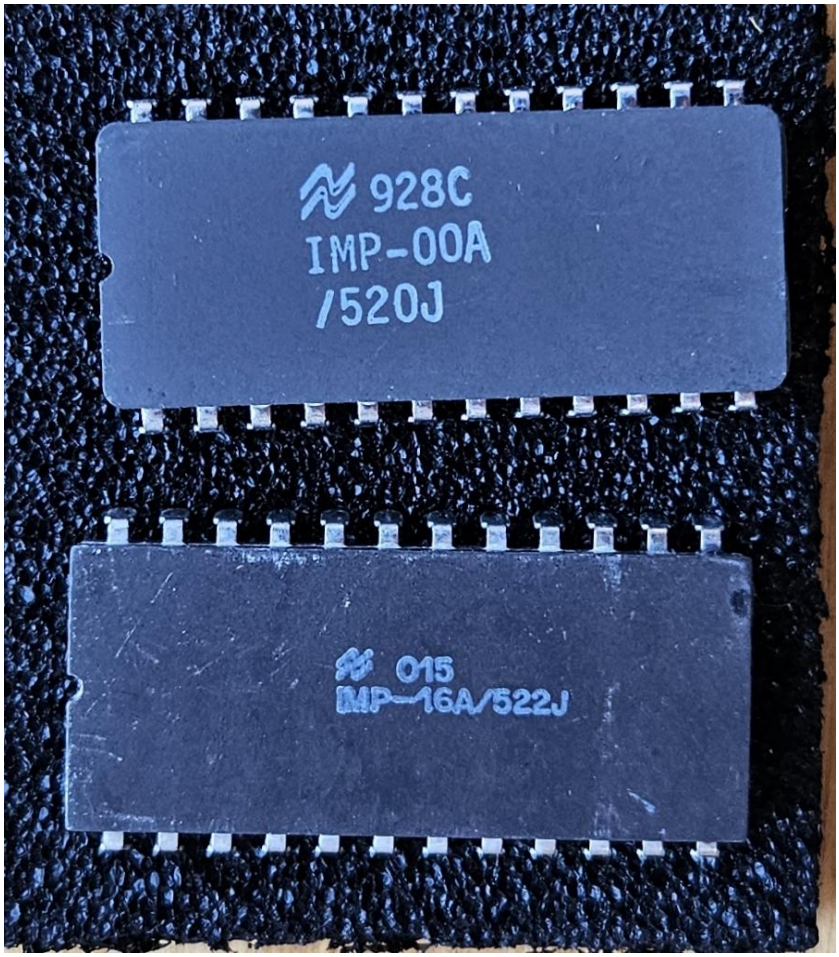


Figure 10: Top: NatSemi IMP-16 IMP-00A/520J register / arithmetic logic unit (RALU). Bottom: NatSemi IMP-16 IMP-16A/522J arithmetic control ROM (CROM).

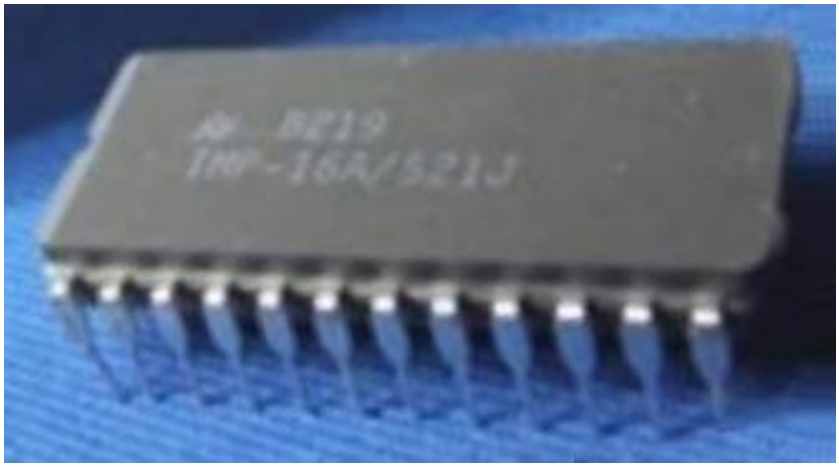


Figure 11: Top: NatSemi IMP-16 IMP-16A/521J standard control ROM (CROM).

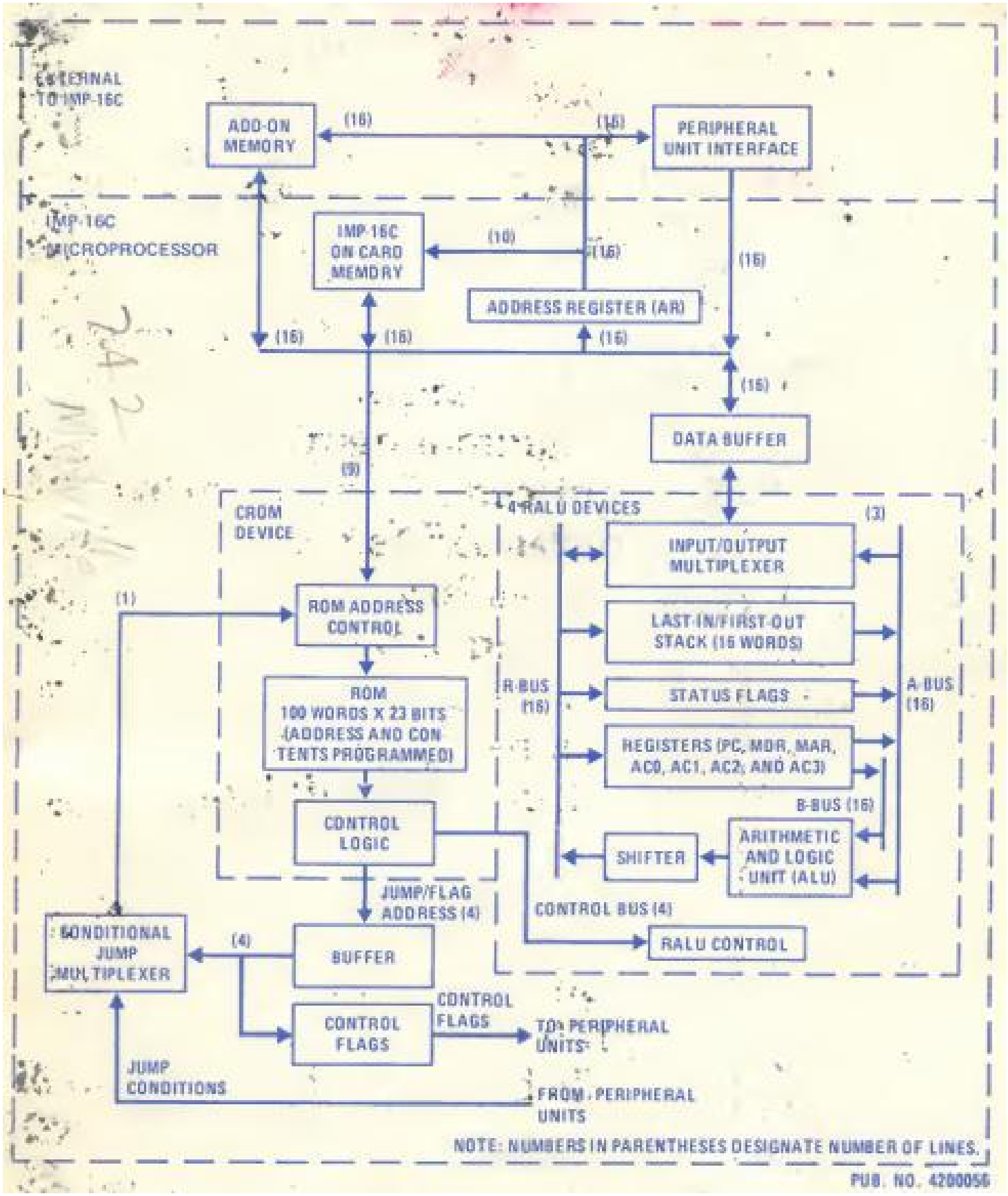


Figure 12: NatSemi IMP-16C computer architecture.

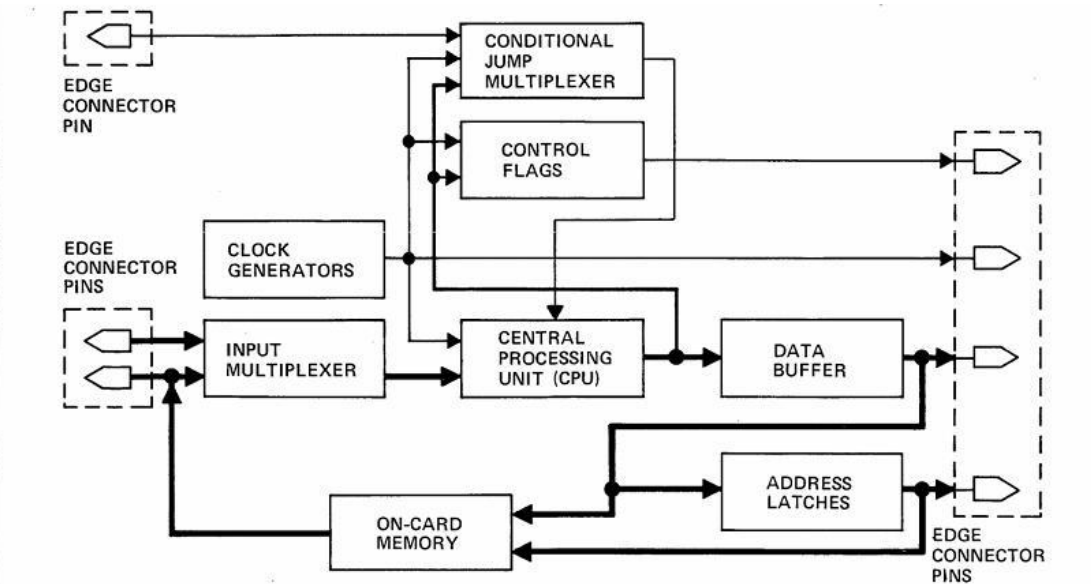


Figure 13: Simplified NatSemi IMP-16C CPU board architecture.

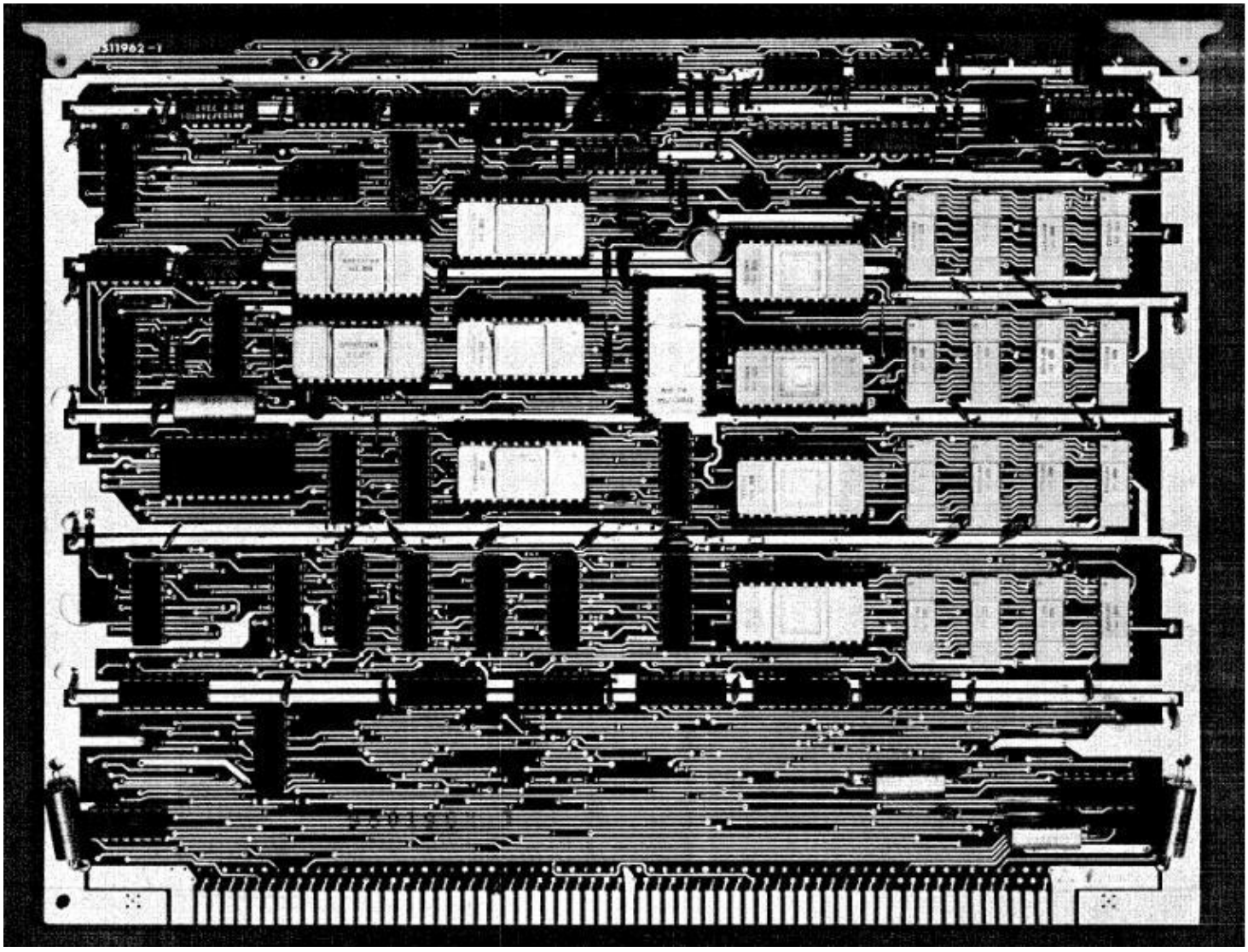


Figure 14: NatSemi IMP-16C CPU board.

Figure 15: NatSemi IMP-16L computer.

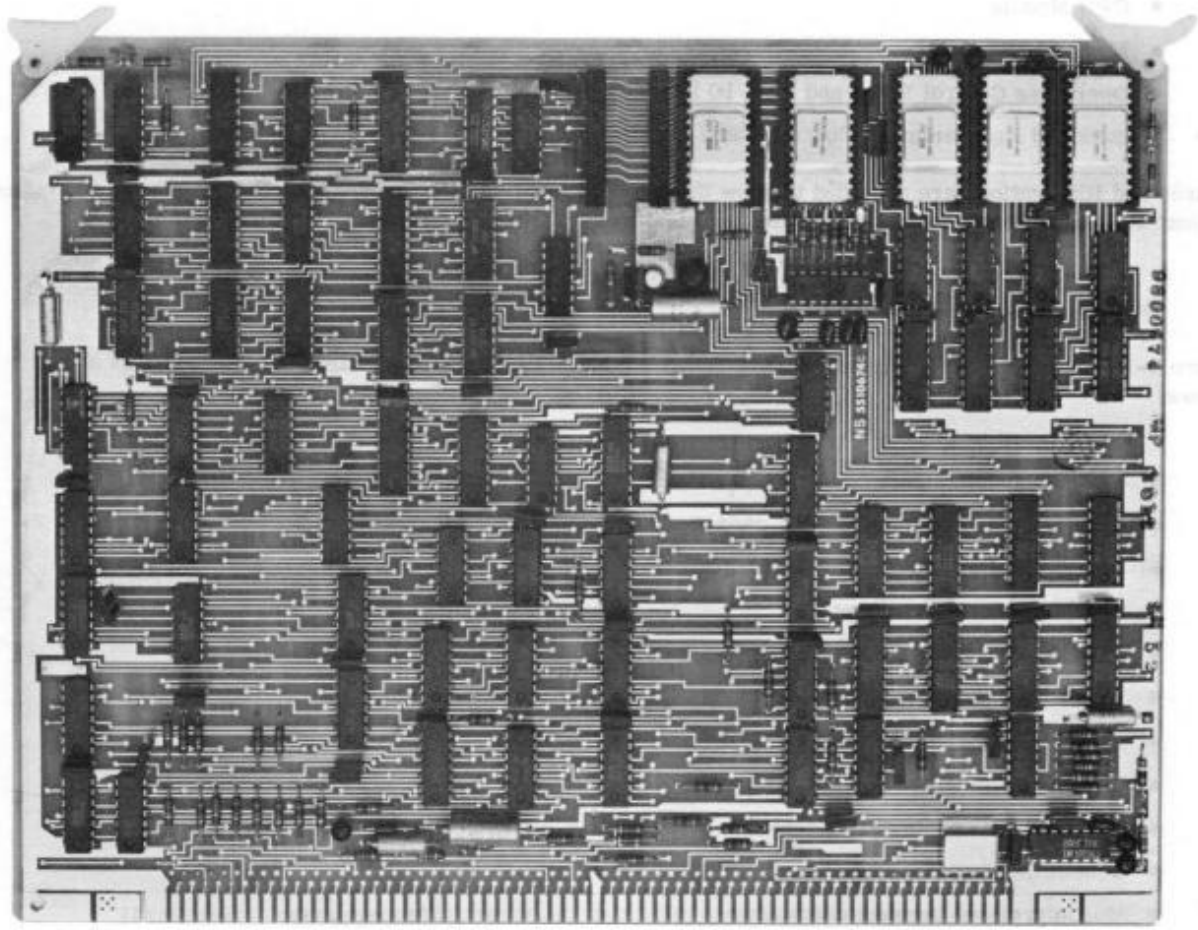


Figure 16: NatSemi IMP-16L CPU board.

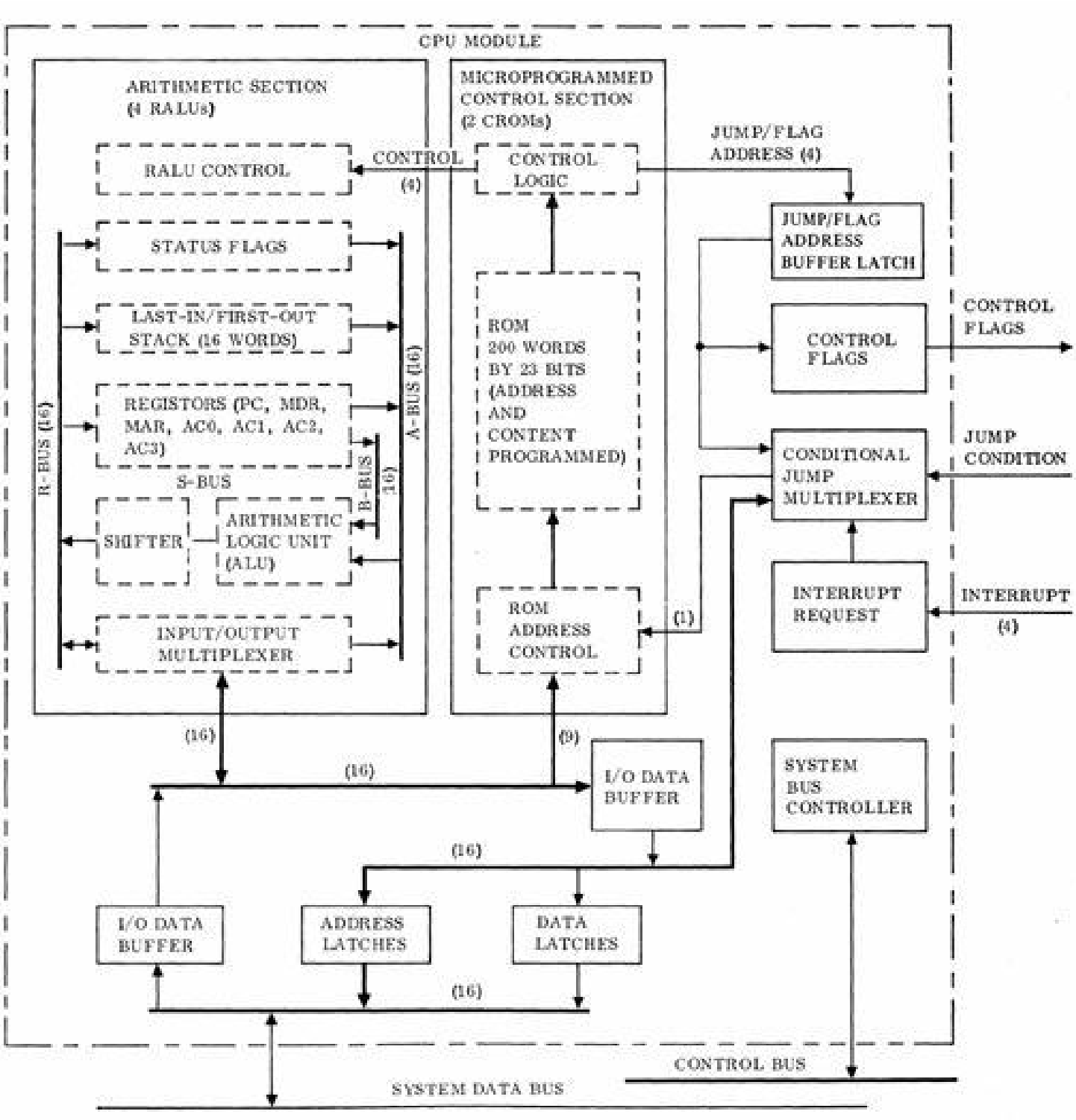


Figure 17: NatSemi IMP-16L CPU board architecture.