Ferranti Argus 400 and Process Control Architecture

JOHN STEELE

Outline of presentation

My role in this development

How did Ferranti start with Process Control

Family tree of Argus computers

Ferranti and Integrated Circuits – MicroNor II

Evolution of the Argus 400 - one of the first integrated Circuit computers

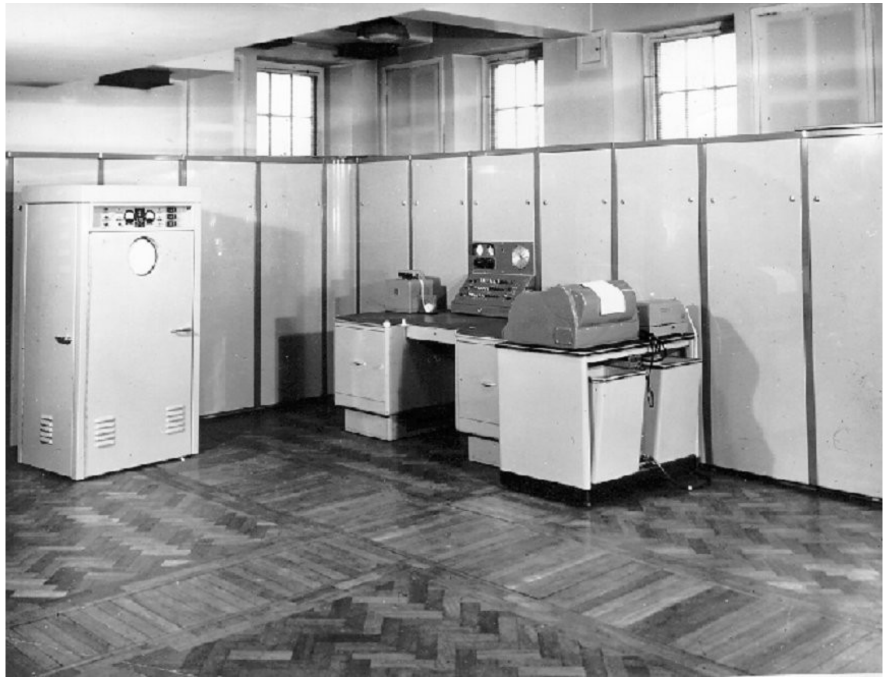
Development of a modular approach to I/O to support Industrial Process control requirements

Some challenges that Ferranti and I faced on the way

Where it all started

I Joined Ferranti in July 1963 with an Electrical Engineering degree from Manchester University

I chose computing as my career in around 1958 on a school trip to see the Mercury computer at the university



My part in the Ferranti developments

I was given the task of the logic design of the Argus 400, my first project, starting in 1963

I contributed to the generation of the standard I/O bus (Interface A)

I designed many of the conventional computer peripheral controllers used on commercial data processing

- Paper tape readers and punches
- Punch card readers and punches
- Various Printer Interfaces
- Magnetic tape (and assisted with aspects of the Disk controller)

I designed several of the other Process Control interface controllers e.g.

- Analogue inputs and outputs, contact inputs, single bit outputs (for relays etc.)
- Panel displays

Background – July 1963

Ferranti Automation Systems Division had been created

The Argus computer (later renamed Argus 200) had been developed as part of the Bloodhound missile program using germanium transistor technology

- One Argus (200) was running at ICL Fleetwood to demonstrate its use for industrial process control
- Another was being installed at West Thurrock Power station to monitor boiler temperatures

A lower cost serial computer named Argus 100 using the Argus 200 technology had been designed and the first two customer machines were being commissioned

- Jodrell Bank (for Mk2 telescope)
- Steel Peach and Tozer (arc furnace control)

A decision had been made to develop a new serial computer, the Argus 400, as a proving ground for the new Micronor II integrated circuits

A parallel computer, Argus 300, was being designed at the same time

Ferranti Argus family tree
Germanium transistor based computers

Wythenshawe developed the Argus computer (later rebranded Argus 200) for Bloodhound program derived from Pegasus instruction set

- Serial/parallel 24 bit instruction, 12 bit data, pegboard program store, core store for data

Argus 100 computer derived from Argus 200

- Serial 24 bit instruction and data all held in core store
- 4096 I/O address space, 4096, 8192 or 12884 words of 24 bit program and data in 4 µs core store

Argus 300 computer derived from Argus 100

- Parallel version of Argus 100 (Argus 350 had some further features)

Ferranti Argus family tree
Integrated Circuit Based computers

Argus 400 based directly on Argus 100

- Serial 24 bit computer, all data held in core store (2 microsecond)
- 4096 I/O address space, 4096, 8192 or 12884 works of 24 bit program and data in 2 µs core store
- Some minor enhancements to instruction set

Argus 500 based directly on Argus 300 with compatible enhancements to instruction set

- Parallel 24 bit computer, all data held in core store
- 4096 I/O address space, with up to 4 banks of core store, each 8192 24 bit words
  - 2 µs and 1 µs variants

Argus 600 – new 8 bit architecture

- Could access a subset of same I/O devices

Argus 700 – new 16 bit architecture

- Could access many existing IO devices

Development of Argus 400

Integrated Circuit Design

Logic design – based on Argus 100 but with significant detailed changes related to the different characteristics of the logic circuits

Mechanical design, PCB layout and manufacture

- Three to five hinged sections interconnected with a flexible PCB
- I/O Section consisting on 18 PCBs with up to 26 TO5 cans on each supporting
  - Interface A, Control console, Primary Tape Input
- CPU consisting of 18 PCBs with up to 39 TO5 cans on each
- One to three core store modules each providing 4096 * 24 bit words

PCB Multilayer printed circuit cards

Interface to modular I/O Architecture and associated Process Control interfaces

Integrated Circuits

Ferranti tended to do everything in-house including IC development

Prior to July 1963 Ferranti had created a DTL range – MicroNor I

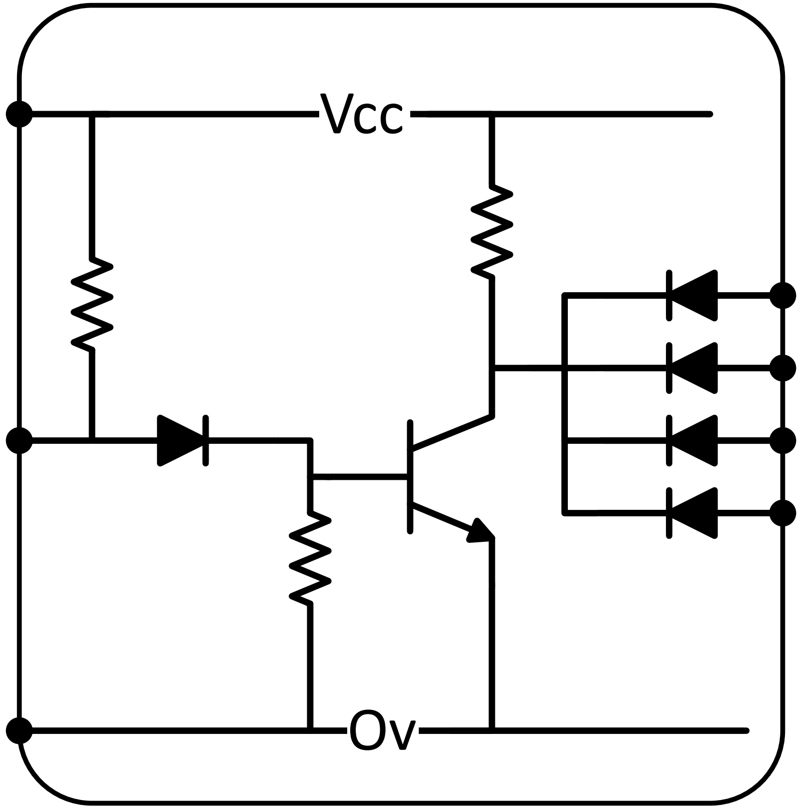
- Strange and unconventional design
- Poor noise immunity

During 1963 a new DTL IC design was created more suitable for computing

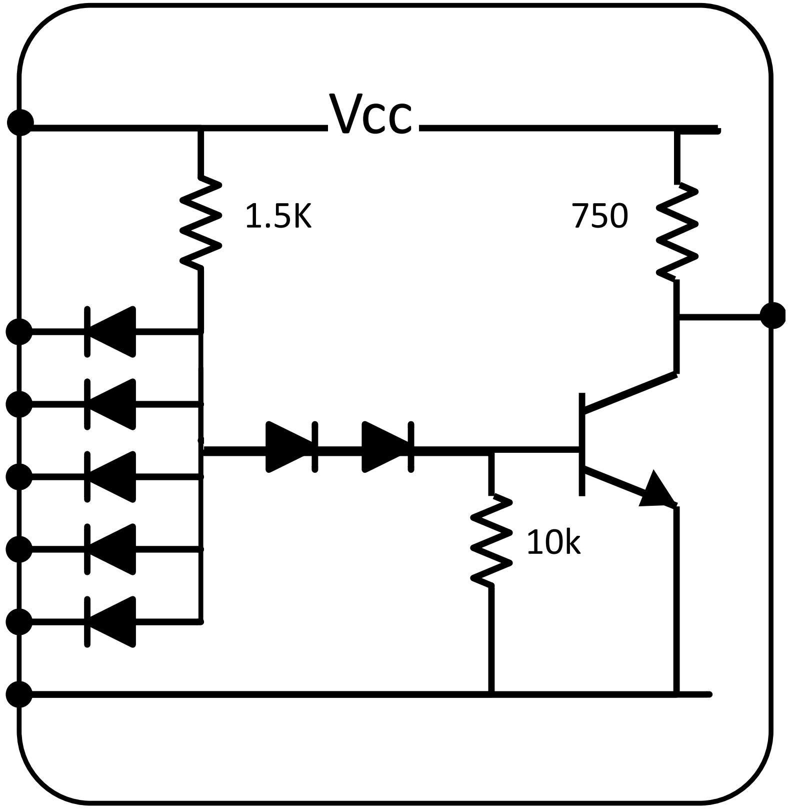
- MicroNor II was already being prototyped when I joined Ferranti

MicroNor I and MicroNor II

Outline approximation - equivalent circuits



MicroNor I



MicroNor II

Updated 14 Jun 2025

Signal standards – What is logic 1?

The germanium transistor based Argus 200, 100 (and later 300) used PNP transistors with signals ranging from 0 volts (defined as logic 1) and -6 volts defined as logic 0)

- Rationale – an output circuit connected to a lamp load should light with a logic 1

The MicroNor range of Integrated circuits retained the Ferranti convention 0 volt as logic 1 and +4.5 volt as logic 0

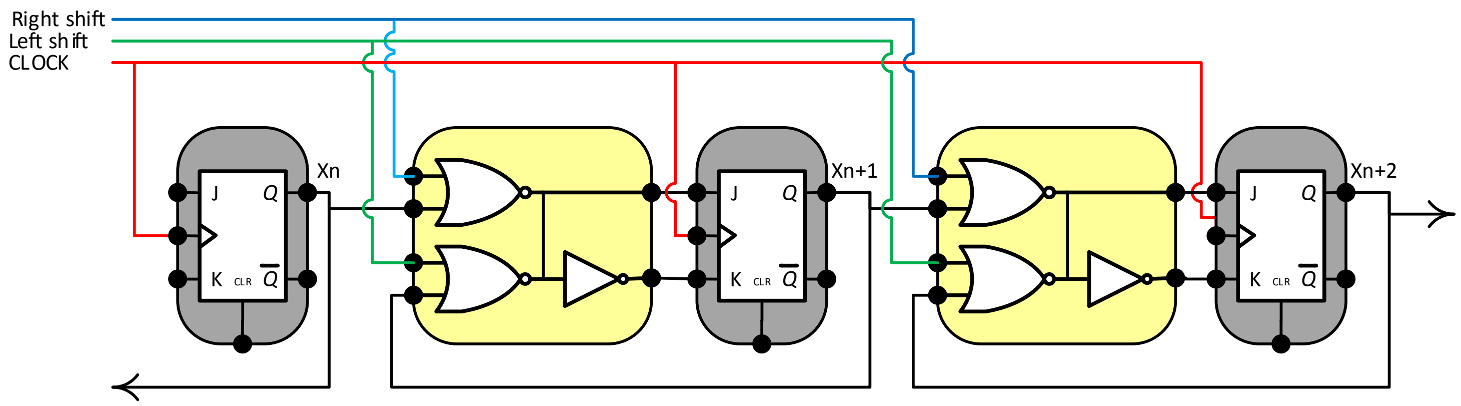
- Unfortunately the rest of the world did not agree!
  - Texas Instruments 74 series NAND gates (released 1966) are actually, in most cases, logically identical to MicroNor 2 NOR gates apart from operating at 5 volt rather than 4.5 volt used by Ferranti.
  - MicroNor II however had significantly better noise immunity.
  - 74 series, being TTL, did not support the "wired OR" function that could be used with MicroNor gates.

MicroNor II characteristics

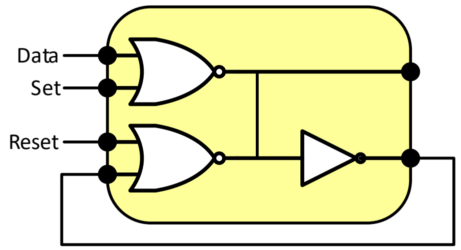
Micronor II was being prototyped when I joined Ferranti in July 1963

- Metal 8 lead TO5 cans (Later packed in 14 lead DIL format to be pin compatible with TI 74 series)
- Good noise immunity - better than 0.8 volt
- Nominal 7 ns propagation delay, maximum 10ns (from memory)
- Fanout of 4 for gates and 3 for bistables
  - Changed later to 8 for gates and 6 for bistables following feedback from Argus 400 design
- Minor SSI to reduce Argus 400 package count
- Power OR gate with fanout of 25
- Temperature range 0 to 50°C (military -20 to 70°C from memory)
- Ferranti retrospectively changed the supply voltage specification from 4.5 to 5 volt (+/-0.25 volt for commercial range, +/-0.5 volt for military range) following introduction of 74 series
  - No chips failed the revised test specification with the change and as far as I am aware no component failures were reported from computers already in service.

MicroNor II- SSI element



Original purpose



Use as a storage element

Compared with 74 series

Noise immunity at least 0.8 volts compared with 0.4 volts

Propagation delay 7ns vs 10 ns and tighter tolerance on maximum delay

No power surge from totem pole circuitry (although rise time was slower)

- Minimal power decoupling required on boards

Fanouts comparable with 74 series for production units

- Power gate was very forgiving - not sure if there was an equivalent 74 series

Voltage specification was changed to be compatible

Wired OR functionality not available for TTL

Dual in line versions of Micronor 2 were designed to be pin compatible with 74 series

Mechanical design

CPU and I/O cards were wire wrapped to pins on the backplane

- 2 * 35 pins using U links

Backplanes were wire wrapped to a flexible printed circuit

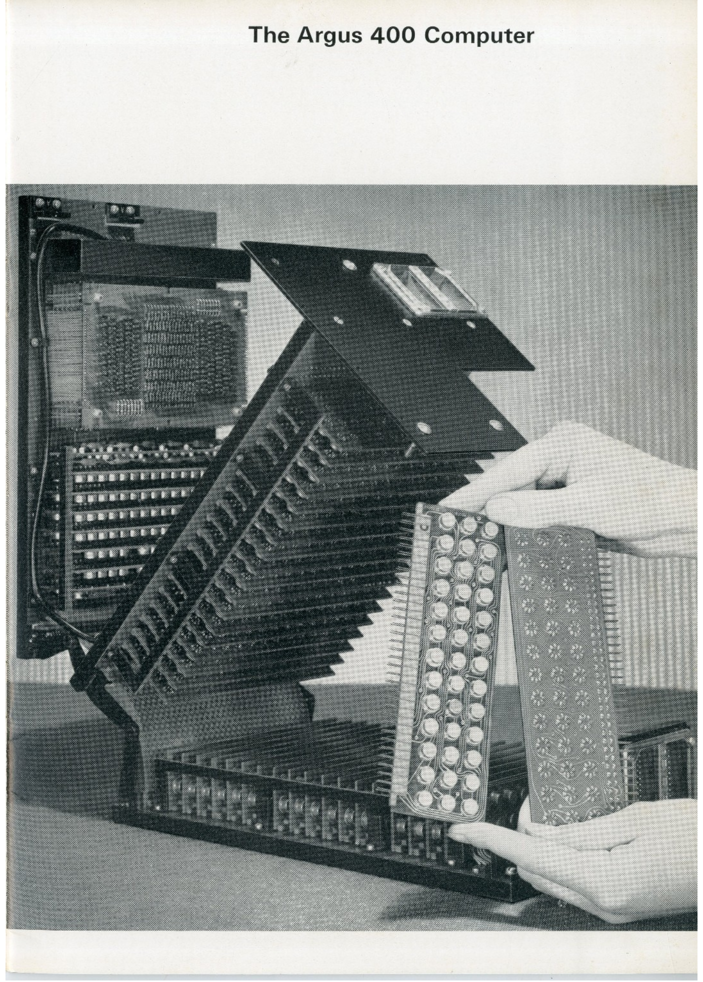
Backplanes were hinged together to allow access for maintenance which led to a problem

- The backplane however was stretched and the midpoint when opening the machine up and tended to break
- We tried, and failed, to find an alternative hinge design that did not stretch the flexible PCB so this was replaced with spring steel strips as an be seen in the TNMoC version

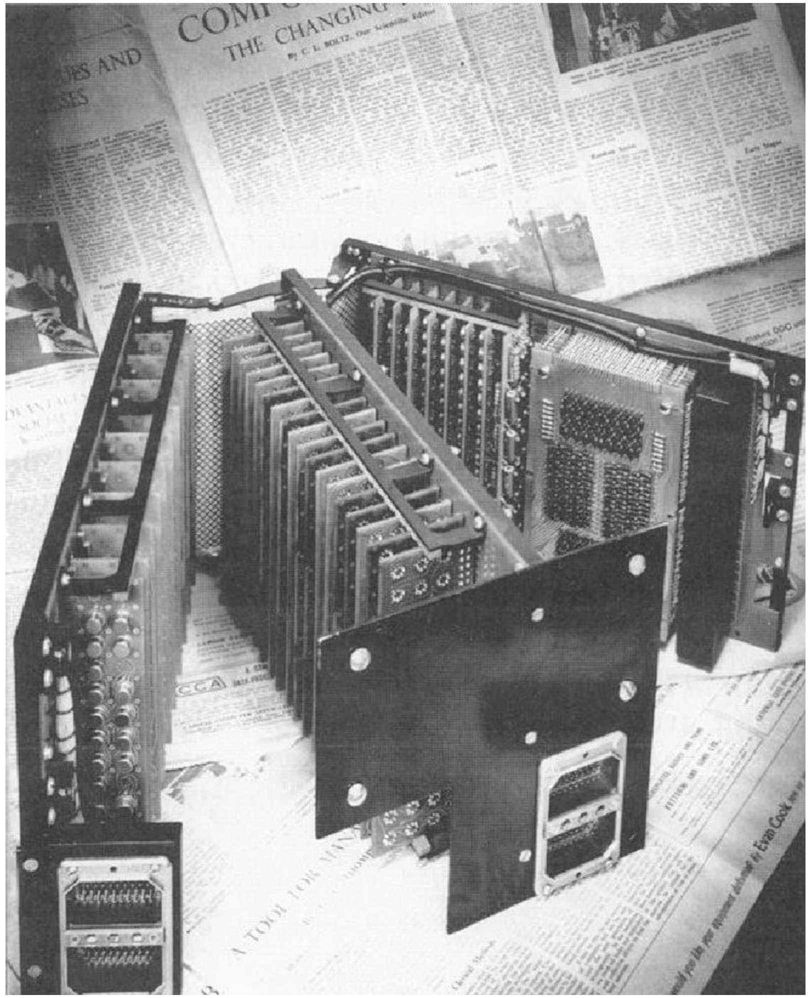
The Argus 400 was intended to be an airborne computer and needed to survive vibration

- Testing on a vibration platform sheared EVERY wire wrap in about 30 seconds
- A better card clamp was successful

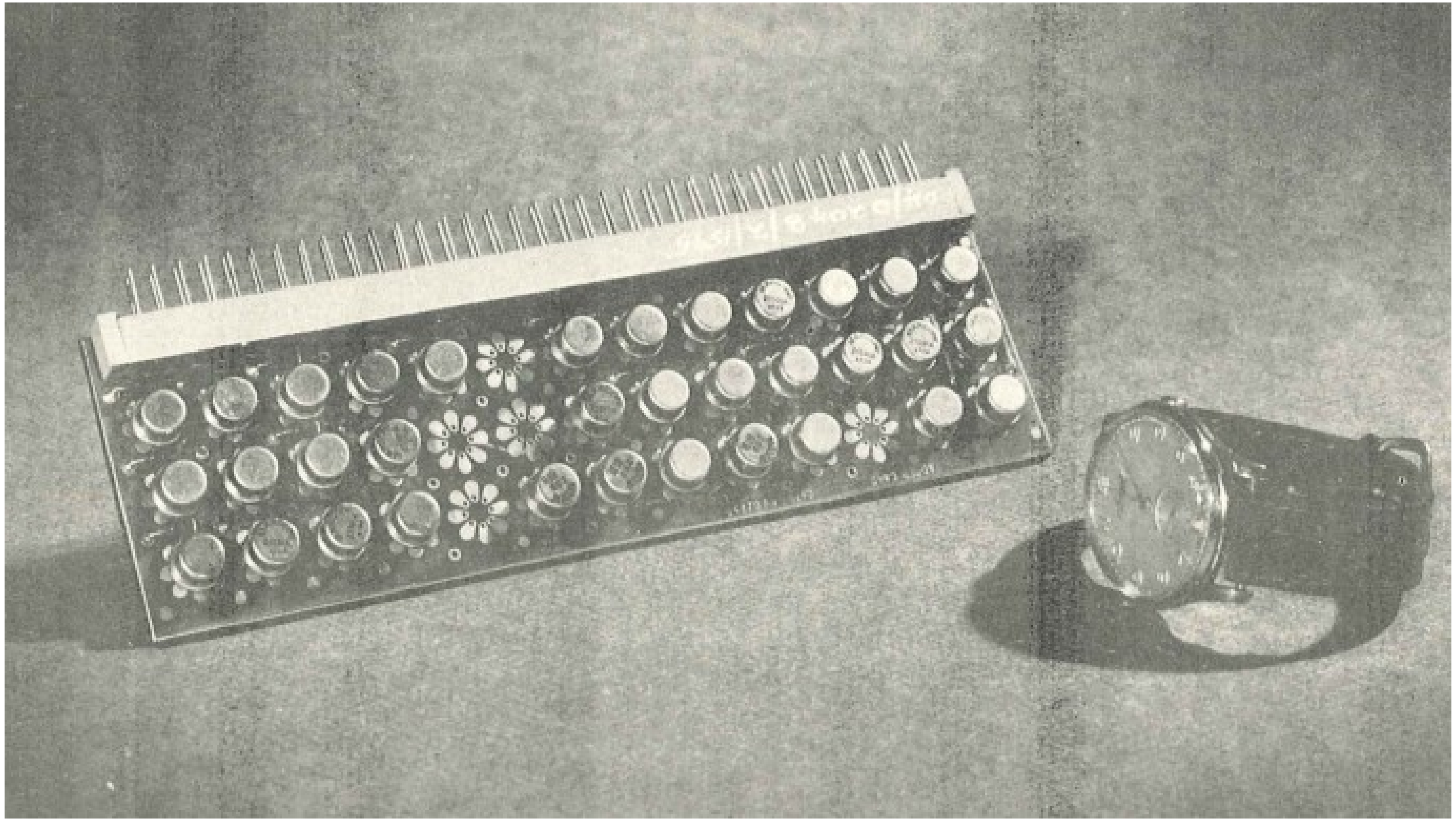
Picture of Argus 400 prototype



The Argus 400 Computer



Example Argus 400 card



PCB design challenges

PCB layouts all done **manually** using drafting tape on acetate film

- One film per layer (about 10 times real size) for the multilayer board
- **Double side boards had been used before but never Multilayer**
- Several card formats were attempted before 18 cards of 39 ICs was chosen for the CPU (26 for the I/O section)

PCB "sandwich" consisted of

- Bottom layer with solder pads and tracks – solder pads limited space available for tracks
- Inner layer with tracks where needed, very close to layer above
- Vcc layer
- 0 volt/ground plane layer very close to Vcc layer (~0.25 mm)
- Upper inner layer with tracks where needed, very close to layer below
- Upper layer with tracks

All holes were "plated through" to connect layers together

Multilayer board manufacture and plating through were all new technology

Argus 400 – requirements (1963)

The Argus 400 was to use Integrated circuits!

- Micronor 1 had been designed (possibly by Maurice Gribble) but had poor noise immunity
- Micronor 2 was being designed (by Peter Bagnall) and the Argus 400 was to be the test bed

The layout was going to use multilayer printed circuits and plated through holes

- This was all to be produced in-house and the processes needed to be developed

Argus 400 needed to be code compatible with the Argus 100

The target clock speed was 4 Mhz (Argus 100 used 500 Khz)

The new core store (designed by Frank Moss) was to have a 2µs cycle time (Argus 100 used 4µs)

The computer was to fit into a standard aircraft instrument case

The environmental specifications were to conform to both commercial (0 to 50C) and military requirements (-30C? to 70C) temperature ranges using appropriate range of ICs

Basic Architecture – Argus 400

24 bit instruction			
14 bit Address 4096 I/O addresses (10000 Octal) 3 * 4096 Memory available	5 bit instruction	3 bit Accumulator X0 – (usually zeros) Maps to memory 10000 X1 to X7 – maps to memory 10001 to 10007	2 bit Modifier X1 to X3

Accumulators (X1 to 7) are actually in defined memory locations (10001 to 10007 octal) and read from memory as needed

- X0 is only written to when reading I/O for compatibility with A100. Usually a source of zeros

Selecting a non zero modifier will add the contents of accumulator to the instruction address before the address is used by the instruction. A modifier adds 3.5 µs to instruction.

Serial arithmetic

- Fewer components than a parallel computer leads to lower cost
- Layout considerably simpler for manual layout on PCB

Supports 8 interrupt lines

Supports 8 “Busy” lines

Supports Direct Memory Access (Direct Store Access) for I/O, pausing computer if necessary, for bulk transfers

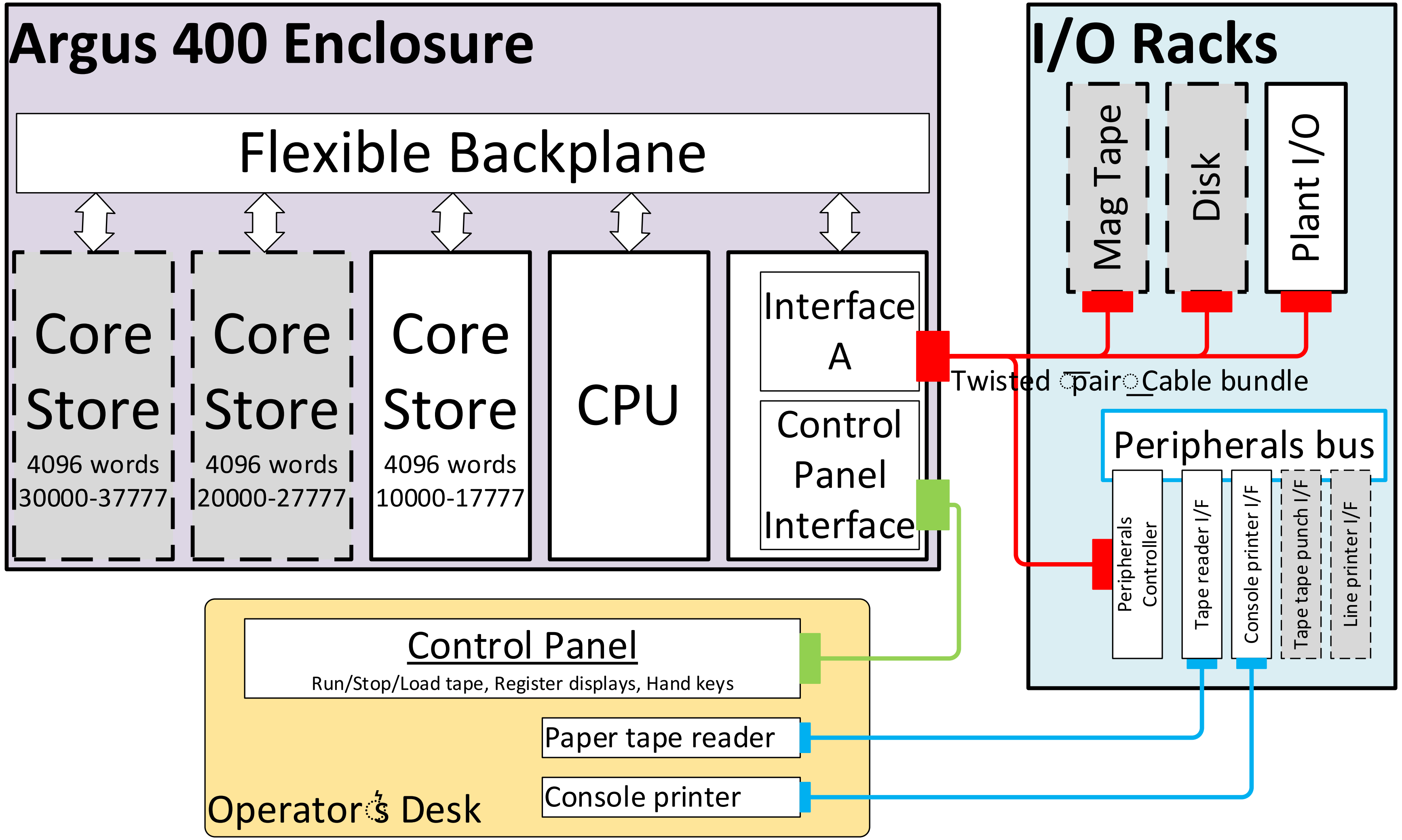
Instruction set (1)

Fn	**Operation**	**Fn**	**Operation**	**Fn**	**Operation**	**Fn**	**Operation**	Reg	Description
00	x’ = n	10	n’ = x	04	x’ = C	14	x’ = n, n’ = x	x	X before operation
01	x’ = -n	11	n’ = -x	05	x’ = -C	15	x’ = x AND n	x’	X after operation
02	x’ = x + n	12	n’ = x + n	06	x’ = x + C	16	x’ = x NEQ n	n	N before operation
03	x’ = x - n	13	n’ = x - n	07	x’ = x +- C	17	x’ = x OR n	N’	N after operation
Jump to N if								q	Q before operation
20	x = 0	21	x ≠ 0	22	x ≥ 0	23	x < 0	Q’	Q after operation
24	Overflow (OVR) set and clear Overflow								
25	Busy line X is set								
27	Unconditional jump to the CONTENTS of N								
Other functions									
26	Used to output data to I/O – Function 10 does same thing in Argus 400								

Instruction set (2)

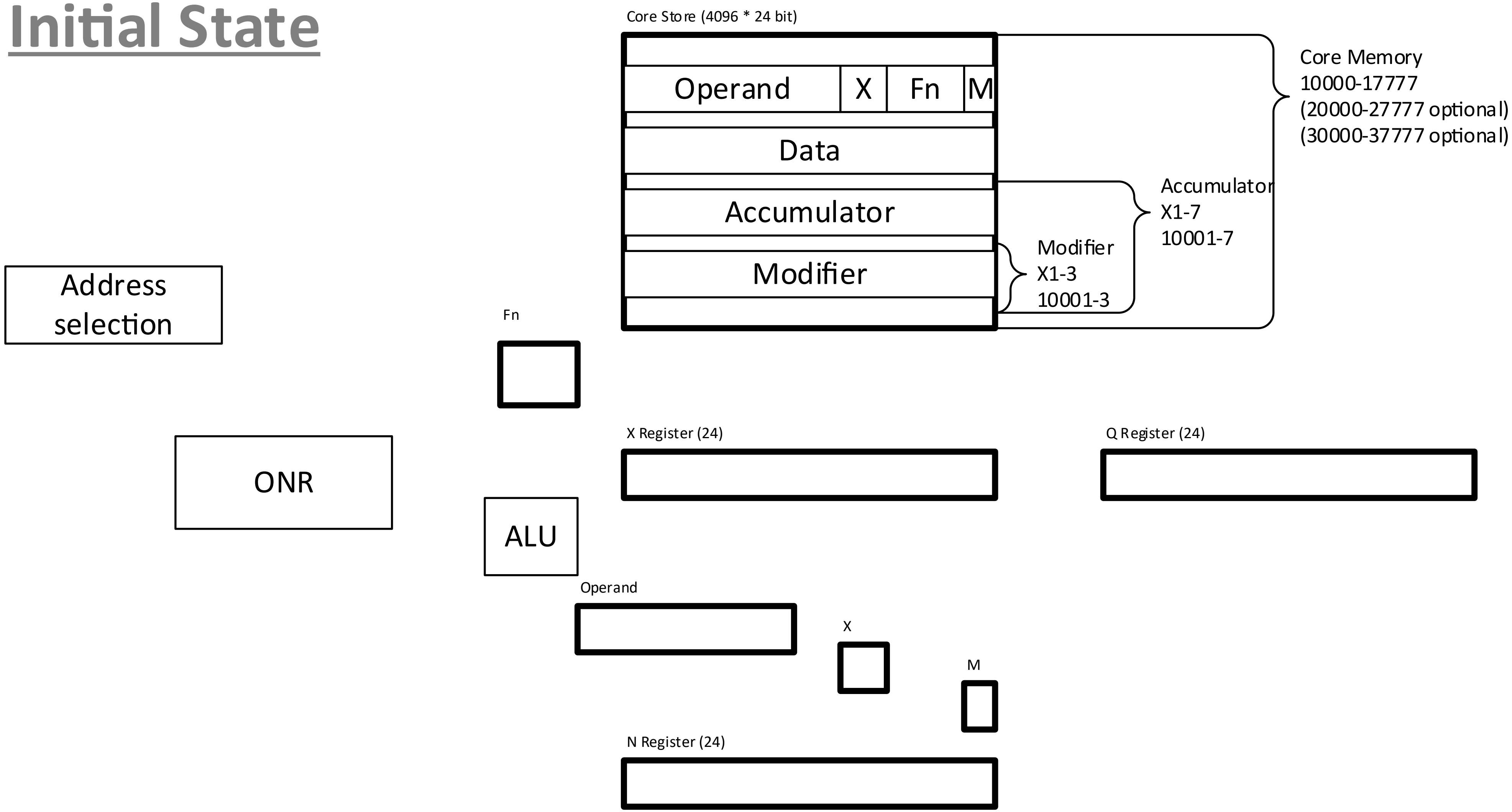
Fn	Operation
30	xq’ = arithmetic shift xq N places right
31	x’ = shift x N places left
32	xq’ = logical shift xq N places right
33	x’ = cyclic shift x N places left
34	Not used in Argus 400
35	Not used in Argus 400
36	xq’ = x * n xq are linked together to form a 48 bit register to receive the product. x and n are treated as signed binary fractions resulting in a signed 47 bit product. The least significant bit of xq is cleared.
37	q’ = quotient, x’ becomes the remainder, resulting from dividing xq by n All numbers are binary fractions

Argus 400 Overview



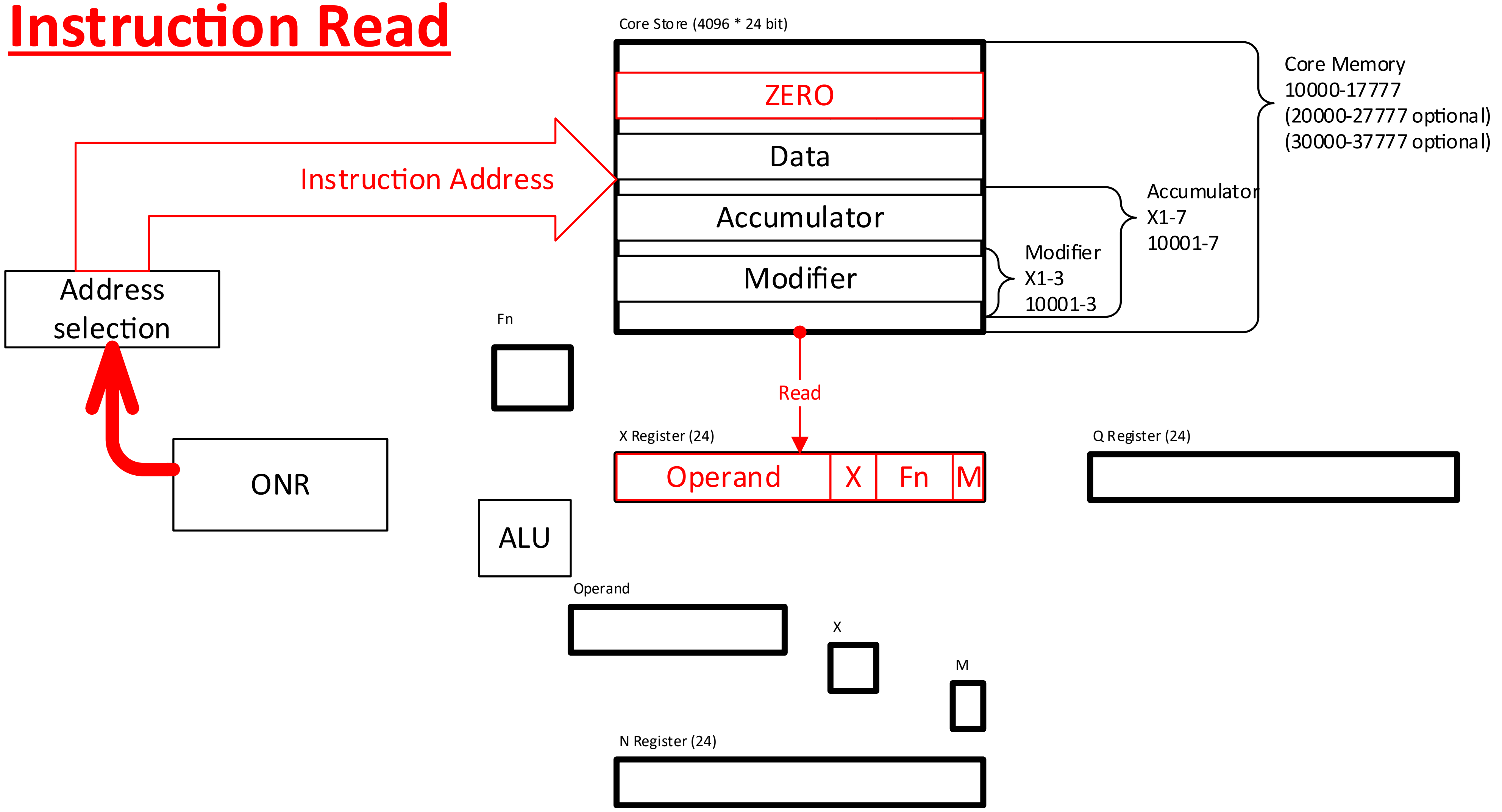
Argus 400 CPU

Initial State



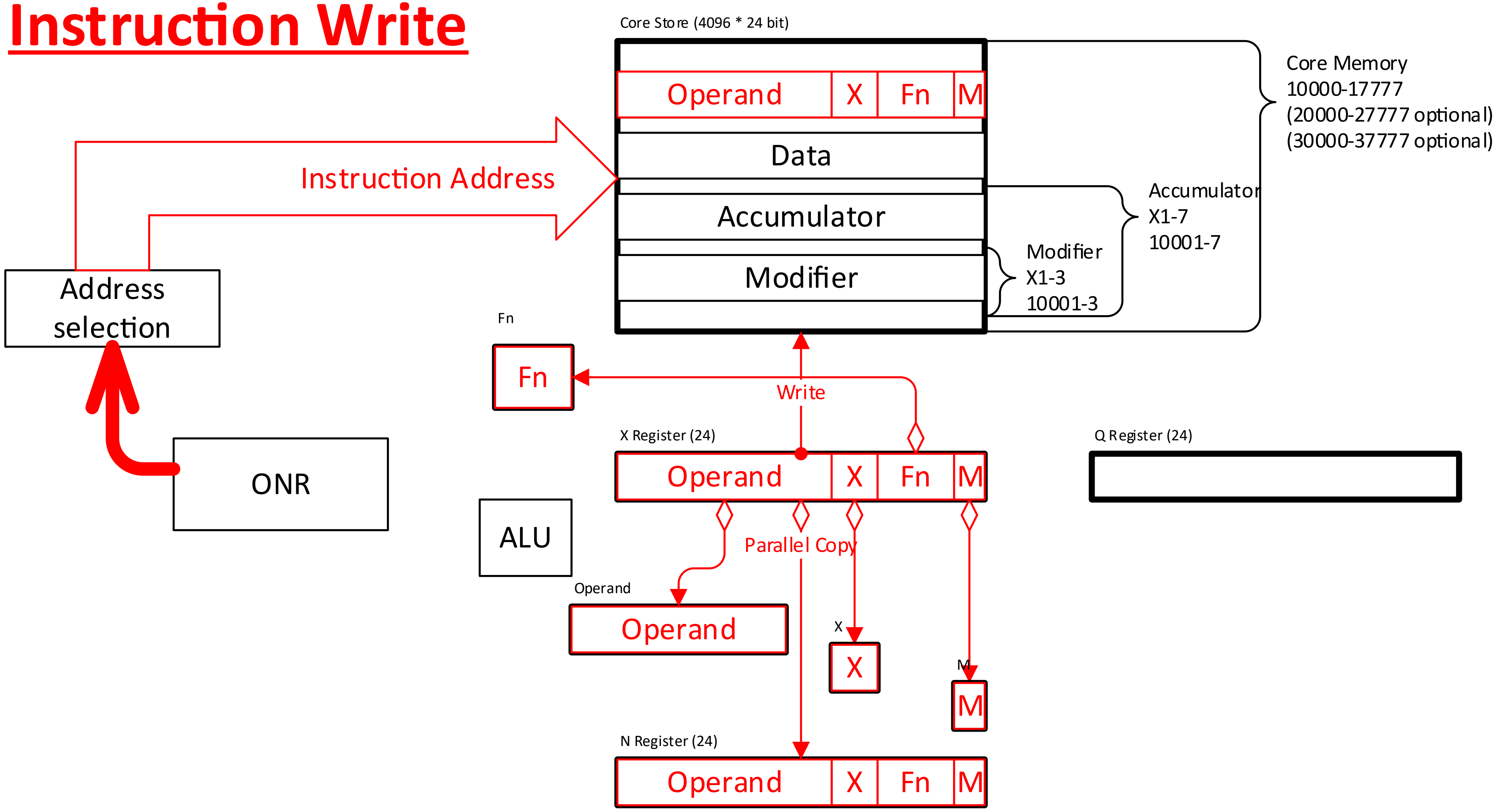
Argus 400 CPU

Instruction Read



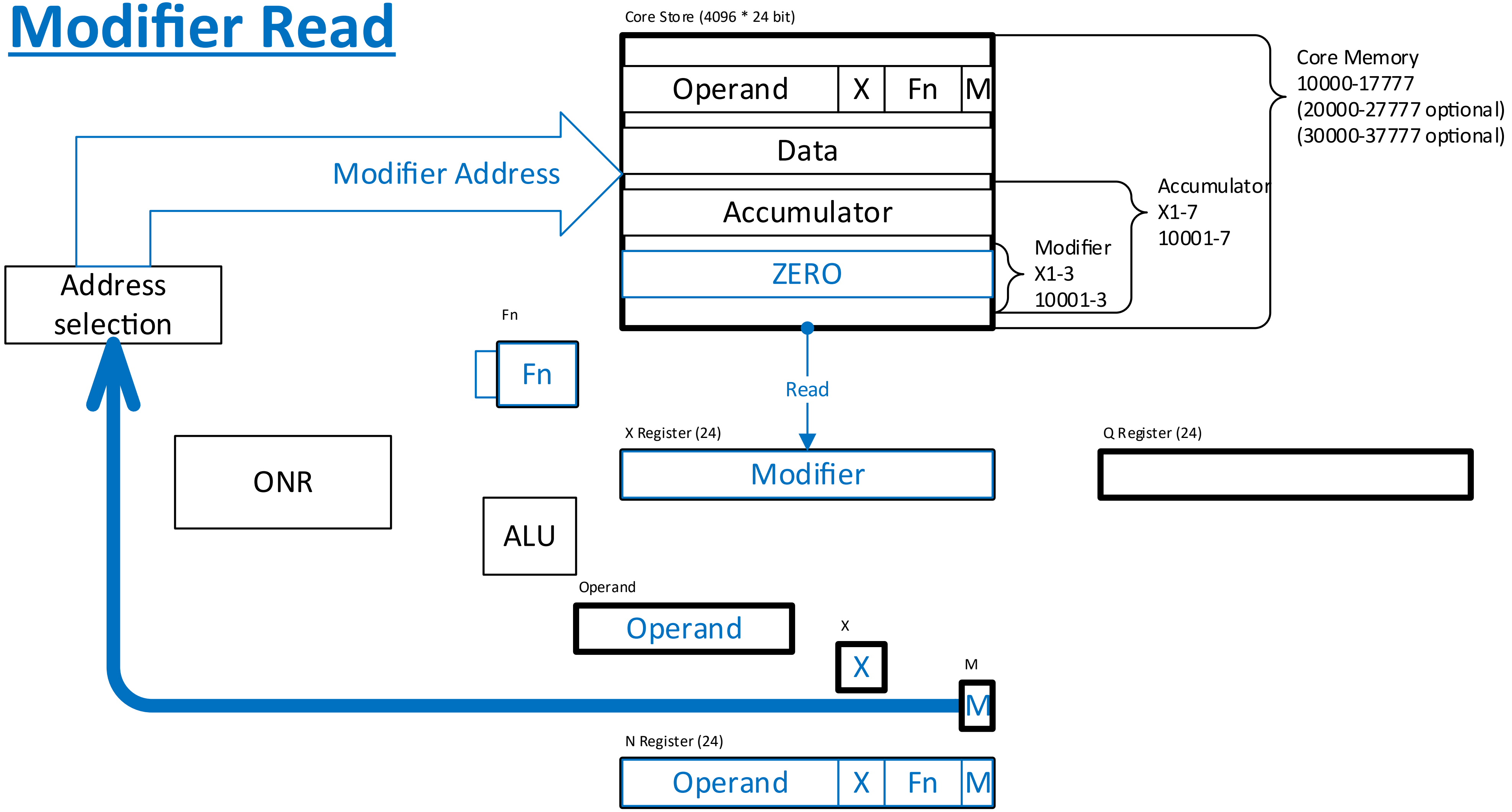
Argus 400 CPU

Instruction Write



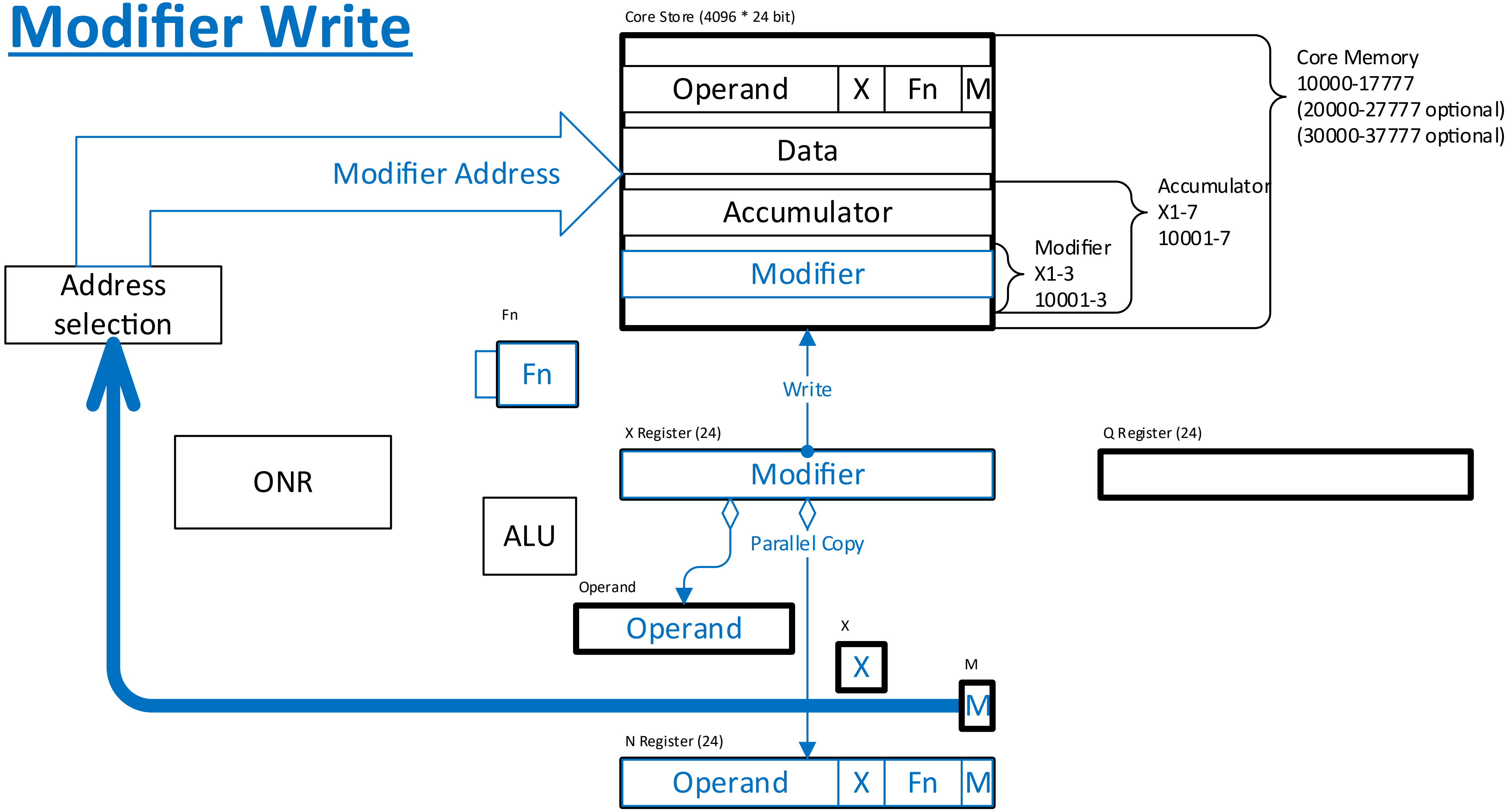
Argus 400 CPU

Modifier Read



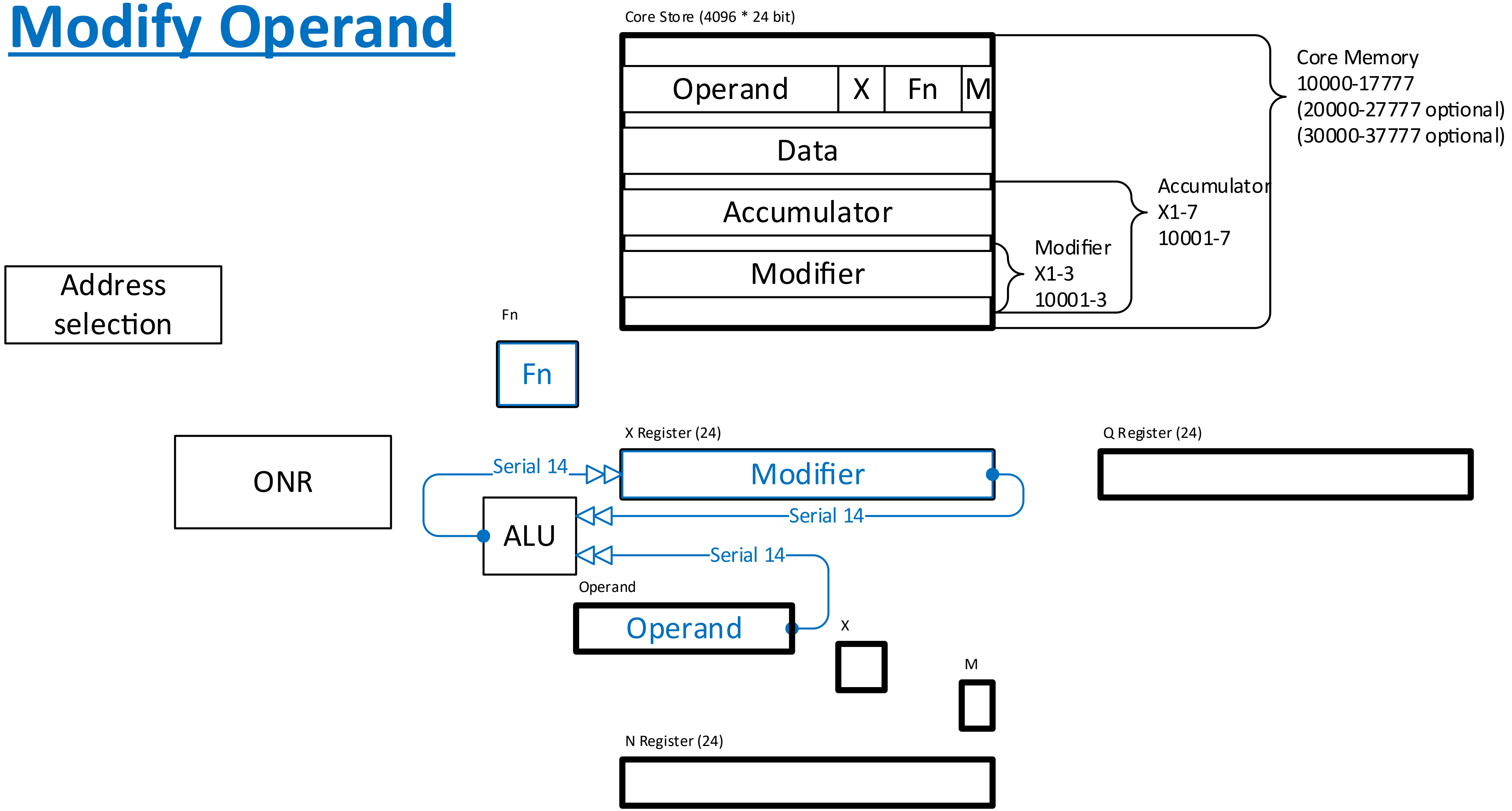
Argus 400 CPU

Modifier Write



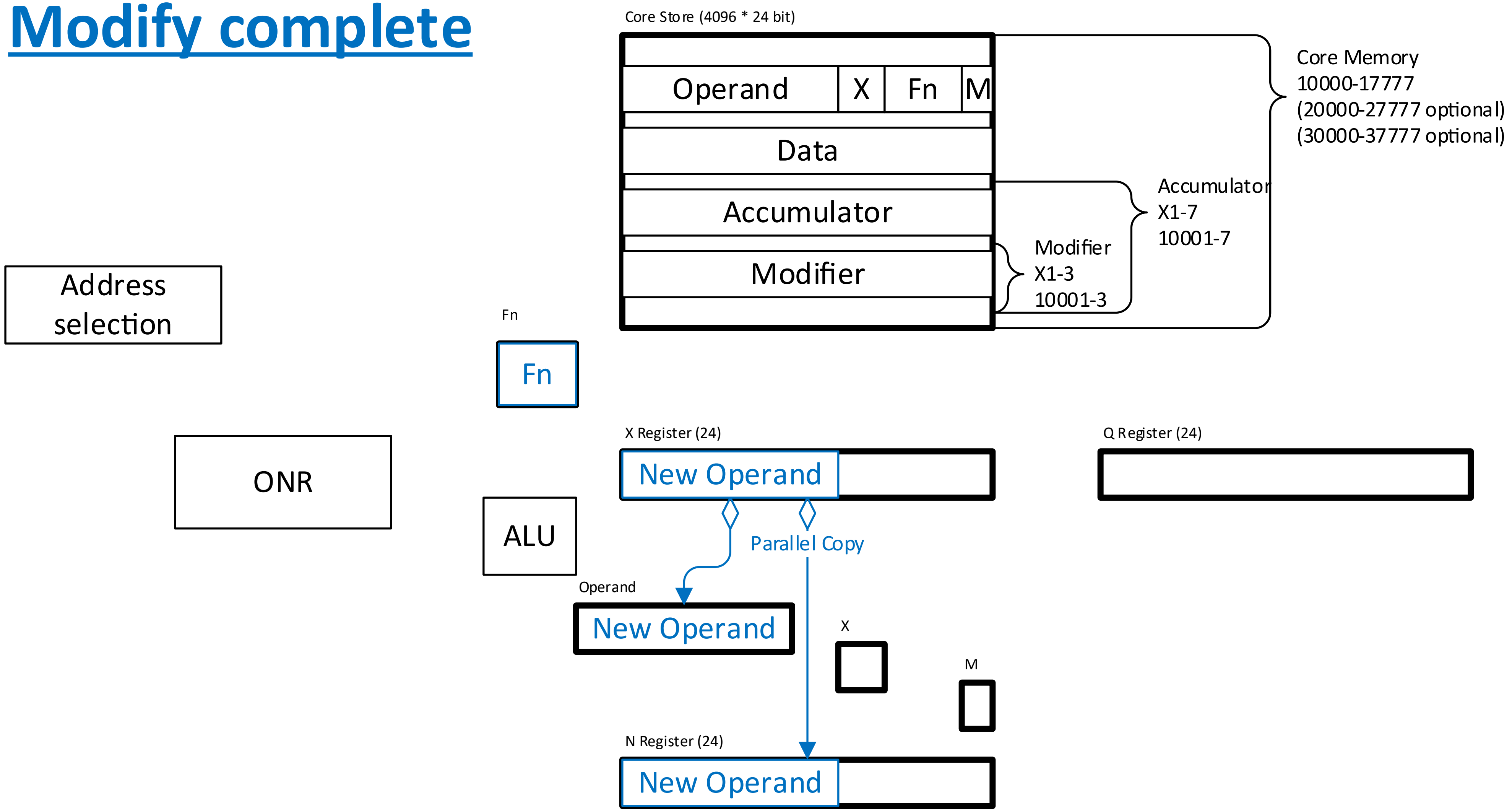
Argus 400 CPU

Modify Operand



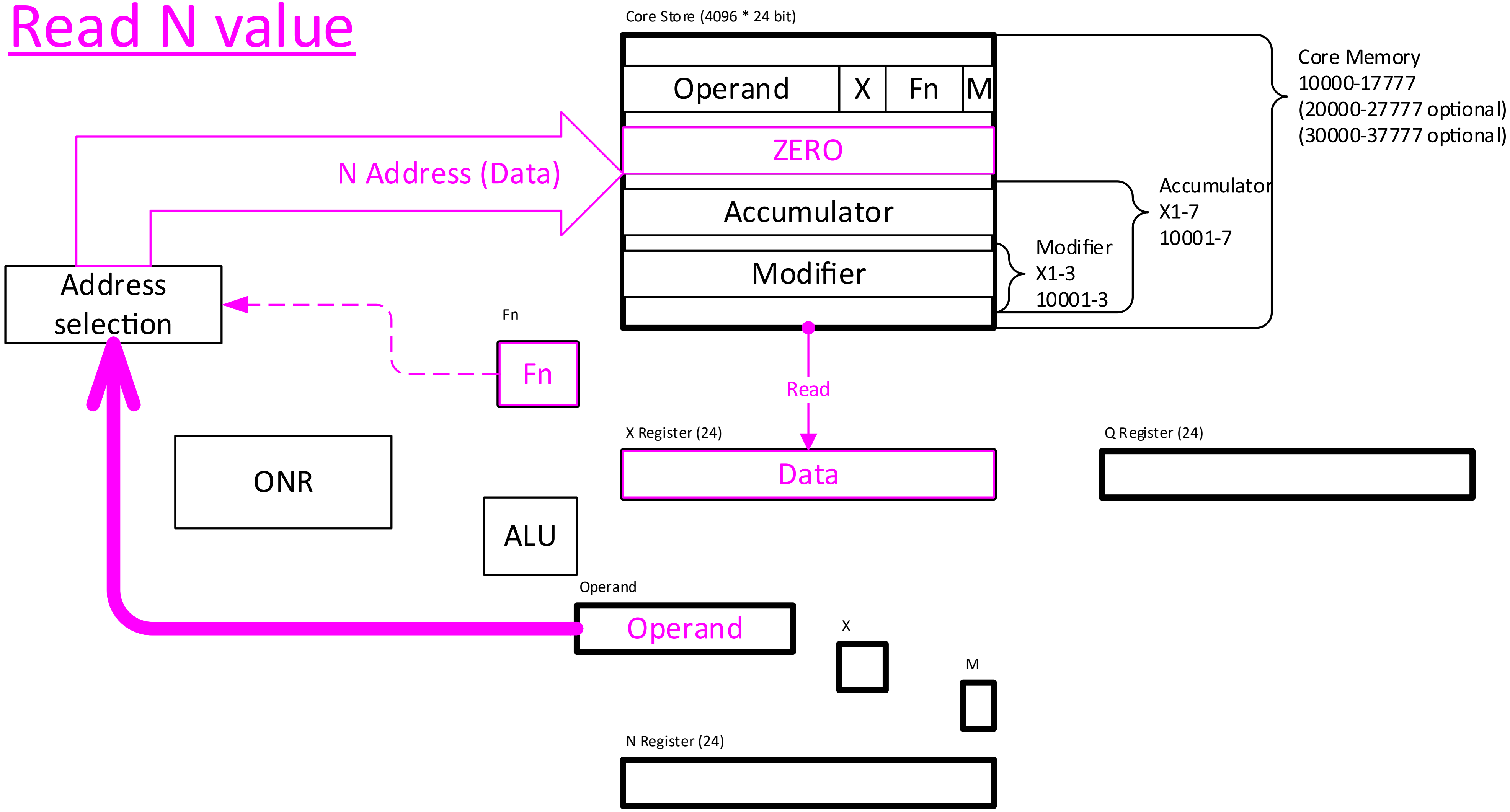
Argus 400 CPU

Modify complete



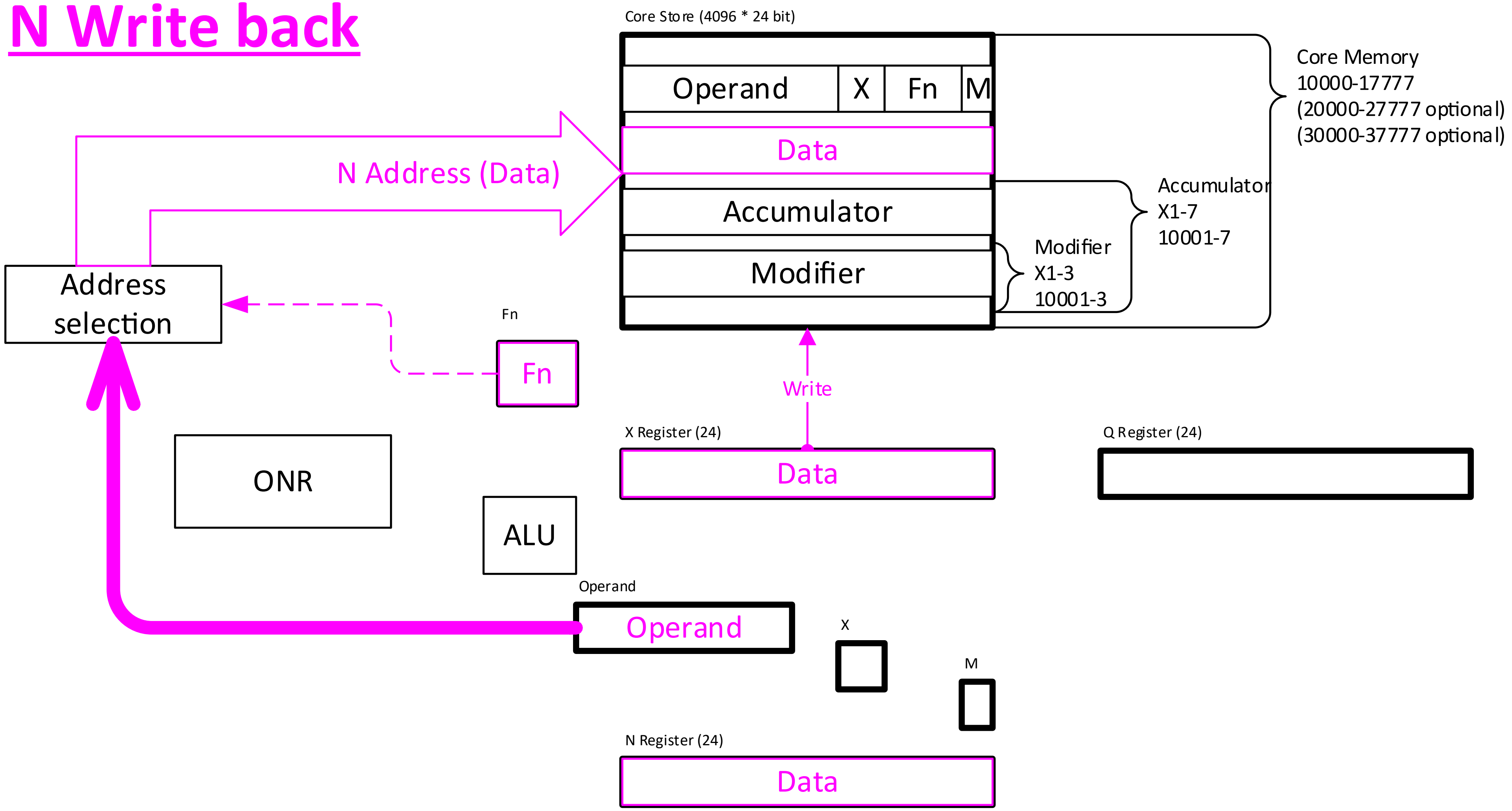
Argus 400 CPU

Read N value



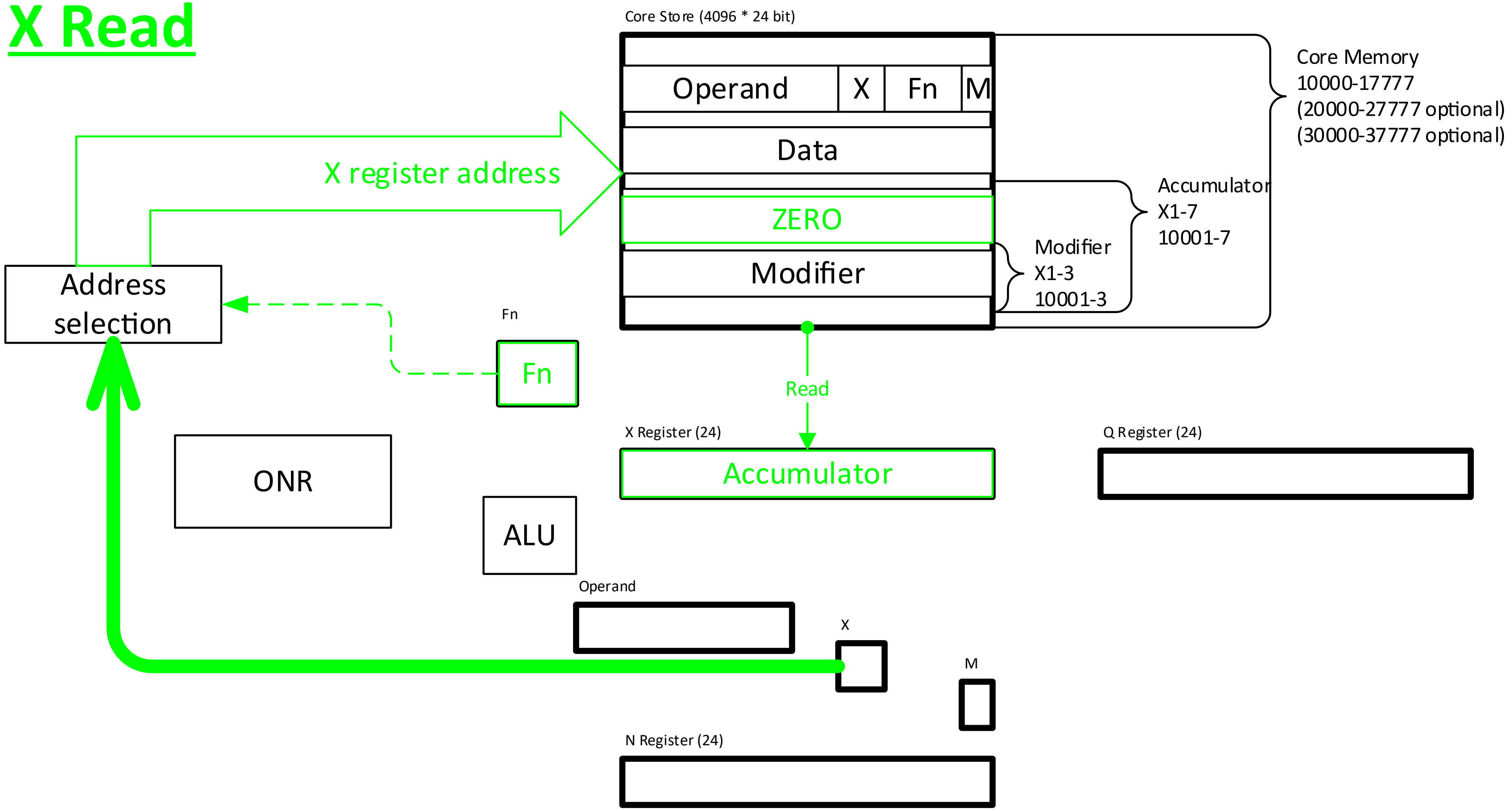
Argus 400 CPU

N Write back



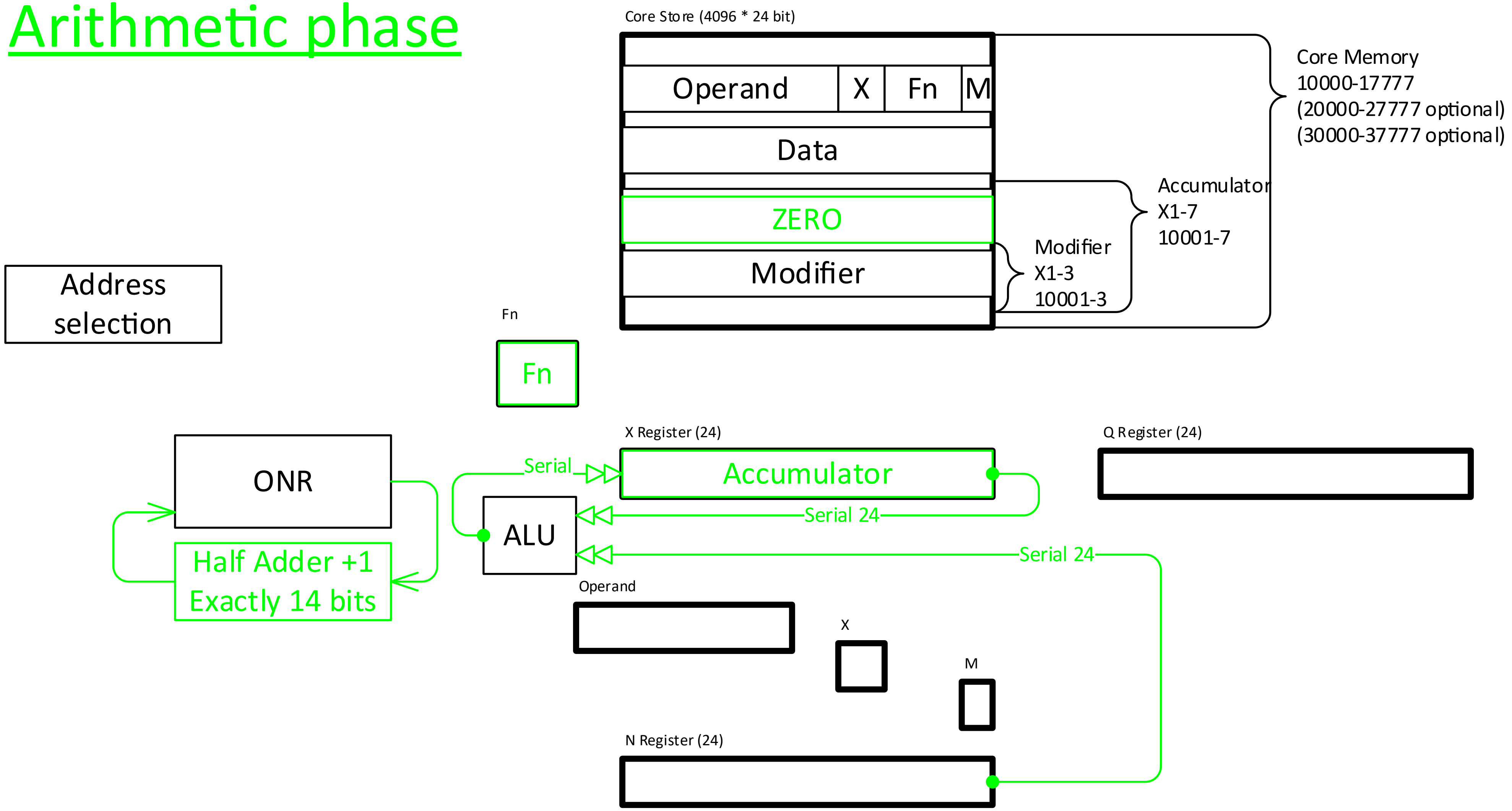
Argus 400 CPU

X Read



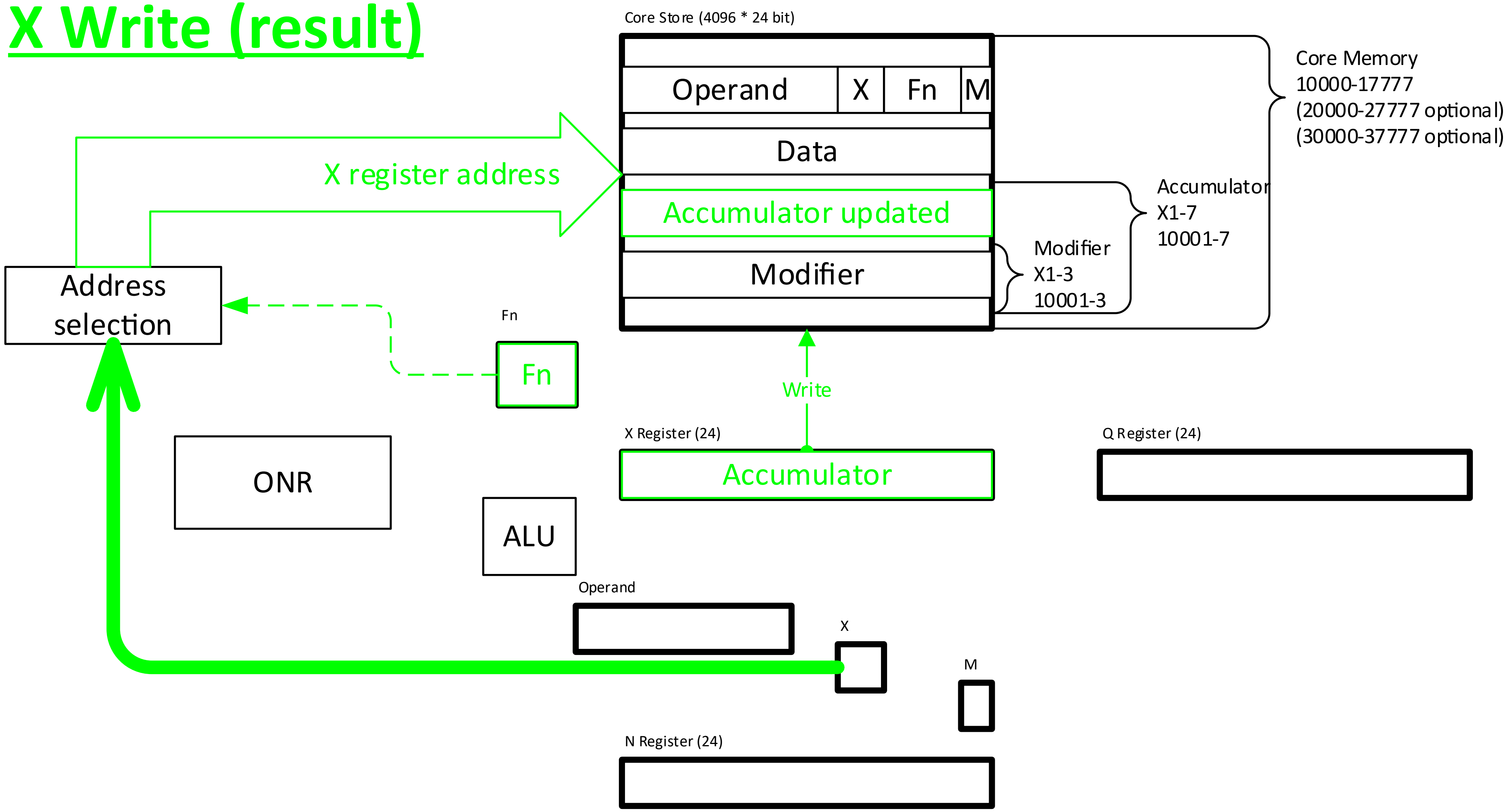
Argus 400 CPU

Arithmetic phase



Argus 400 CPU

X Write (result)

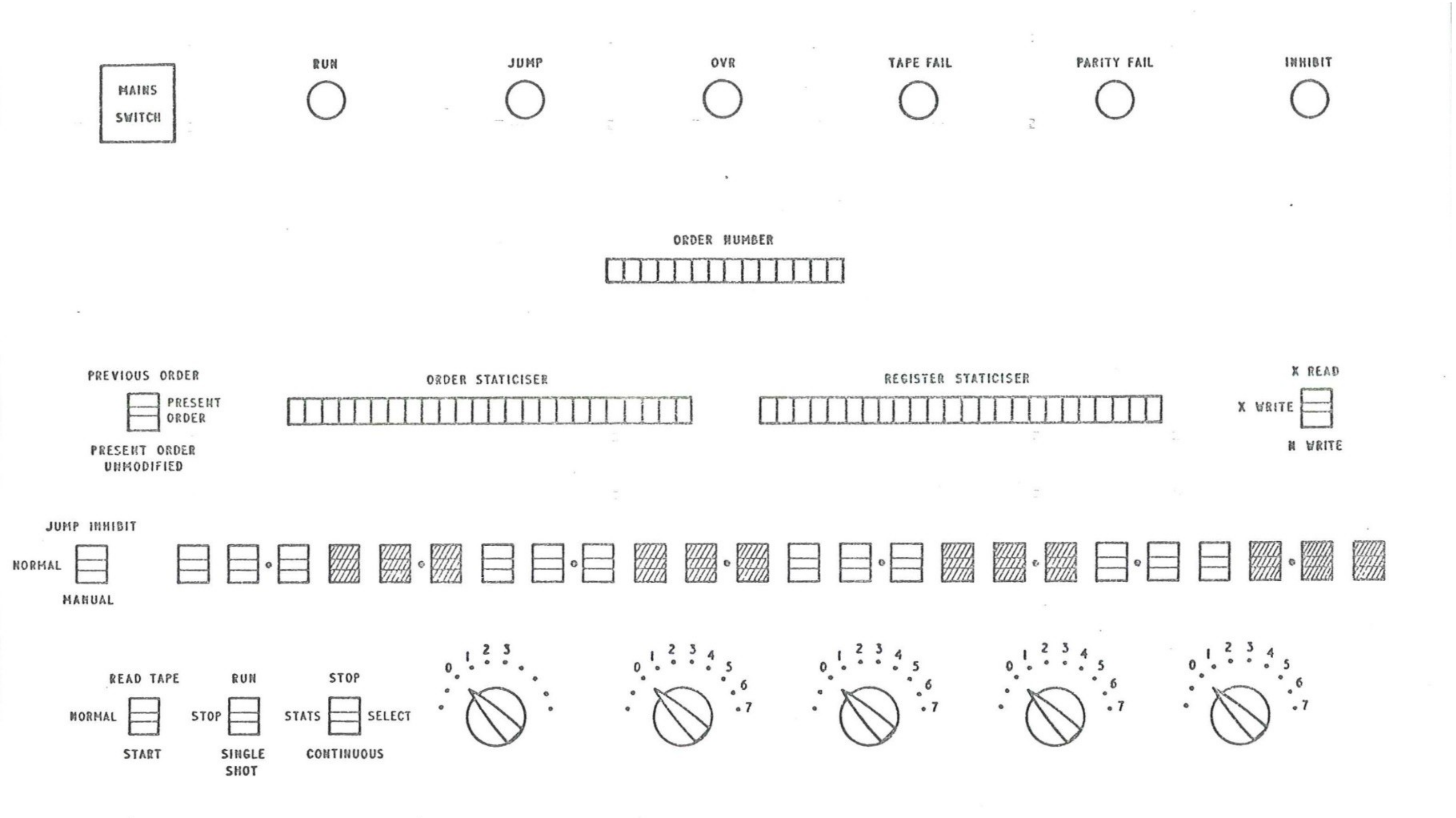


Instruction timings

Function group	Memory accesses Inst r, Inst w +	Serial clocks	Instruction times	Notes
Modify	Mr, Mw (optional)	14	2 + 3.5 = 5.5 μs	Optional modification – add to below
00-03, 10-17	Nr, Nw, Xr…Xw	24	6 + 6 = 12 μs	X and N may be interchanged for some function
14	Nr, Xr, Nw, Xw	24	6 + 6 = 12 μs	Two successive reads to allow data to be swapped
Shift functions	Xr…Xw	0 to 31	4 + 5.5 = 9.5 μs to 4 + 7.75 = 11.75 μs	14 clock cycles required for ONR update
Jumps	Xr…Xw	14	4 + 5.5 = 9.5 μs	
36 (multiply)	Nr, Nw, Xr…Xw	48 to 600	6 + 12 = 18 μs to 6 + 150 = 156 μs	Implements the “Booth algorithm”. Timing depends on data pattern in X.
37 (divide)	Nr, Nw, Xr…Xw	600 to 624	6 + 150 = 156 μs to 6 + 162 = 162 μs	Depending on whether a final restore is needed

Argus Control Panel

Argus 100 version



Interface to I/O – Standard devices

Process Control requires Standard computer peripherals such as

- Standard “Small” Peripherals such as
  - Paper tape reader
  - Paper tape punch
  - Console printer/keyboard (including re-perforator and tape reader
  - Card punch
  - Card reader (developed for Argus 600)
- Bulk storage
  - Rotating disk storage
  - Burroughs fixed disk - dedicated controller
    - 1 Mbyte storage, head per track
  - Magnetic Tape – 1 to 4 tape drives managed by one controller
    - Ampex TR7 drives
    - 36 inches per second
    - NRZ (800 bits per inch)
    - Around 10-15 Mbytes on 2400 ft tape on 10.5 inch spools depending on block size recorded

Interface to I/O – Specialised devices

Analog inputs and Outputs

Analog inputs e.g. from thermocouples measuring boiler temperatures

- Typical many hundreds of such data sources on a system
- Analog to Digital converters needed to convert inputs to computer readable values
  - A to D converters Need to be accurate and the Ferranti version had 12 bit resolution ⇨ EXPENSIVE and hence need to be shared
  - Programs need the data in memory and hence it is convenient if the inputs can be automatically scanned sequentially, digitised and and transferred into a corresponding block of memory
  - Each analog device (e.g. thermocouple) needs to be selected sequentially in a cyclic manner and switched to the A/D converter. The voltage levels are low (millivolts) and semiconductor switching there was infeasible. We had:
    - Relays, and both dry reed (rated at 10^8 operations) and mercury wetted reeds reed (rated at 10^9 operations) were used (some sites prohibited any mercury on site)
    - Rotary scanner

Analog outputs

- These tended to be far fewer and (as far as I recall) these were simple D/A converters feeding an analog output of several volts to an external controller

Interface to I/O – Specialised devices

Digital Inputs and Outputs

Single bit inputs e.g. process control panel switches grouped in 24 bit words

- Read by software as required

Single bit outputs

- Indicator lamps on panels (typically 24 volt signal)

Panel Numeric or character displays

- Including Latching displays that would retain value on power failure

Note that the process control panels here were custom designed and should not be confused with the main computer control panel.

Interface Bus

Interface A- Overview

Required to span multiple racks

Requires to have good noise immunity consistent with industrial environment

Required to support a wide range of I/O address space required for Process Control

- Multiple modular controllers

Required to be "universal" for future range of Ferranti computers

- Argus 400
- Argus 500
- Argus 600 (small subset)
- Future Argus computers (I/O devices still used on A700 7 years later)
- Cabling used (at least initially) a "triple twist" cable construction for each signal
  - Better than twisted pair
  - Coax would have been better but more difficult to terminate
- Impedance (150 Ω) was approximated to minimise transmission reflections

Interface Bus

Interface A- signals

Signals	Description
AN1 – AN12, AN8’ – AN12’	Twelve address lines to select I/O component being addressed. The inverted address lines allow simple module level decode using jumpers/DIL switches
ASET	Control line indicating address lanes are valid
DSAI and DSAO	Direct Store Access (DMA) Input request/ output request
DSAA	Computer accepted request and DSA transfer can continue
DSAH	Hold computer for bulk DSA transfers
CSA1’ – CSA16’	Sixteen core store address lines (Argus 400 would only use fourteen) used for DSA (DMA)
DIN1’ – DIN24’	Data input (to computer) lines used for program and DSA (DMA) data inputs
DO1 –DO24	Data output (from computer used for both program or DMA outputs
DSET	Valid data is available on Data Output lines

Interface Bus

Interface A – signals continued

Signals	Description
INTE0 – INTE7	Eight Interrupt lines. Each can have more than one device connected.
0B – 7B	Eight Busy lines. Each can have more than one device connected.
IOLO	Interface lockout – can be set by computer
RNSW	Control panel RUN switch is in the Run state
STSW	Control panel START switch is in the Start position

Example Interface types

Standard peripherals

- Paper tape reader (300, 500 or 1000 characters per second)
- Paper tape punch
- Console printer e.g. Teletype ASR33
- Line printer
- Card reader (specialised system e.g. remote batch terminal)

Data storage

- Disk drive
- IBM compatible Tape Drive

Specialised interfaces

- Analogue inputs – need specialised scanners to feed millivolt inputs (thermocouples) into shared A/D converter
- Analogue outputs
- Digital inputs e.g. control panel buttons/switches
- Control panel displays
- Graphical displays
- Tracker balls
- etc

Some lessons learned along the way

Argus 400 calculated a different value for Pi than the Argus 100 and 300 to 500 places of decimals.

- This took a lot of effort to find the cause! It took over 5 minutes computation before it printed any results.
- Lesson learned – be methodical and patient! It took two weeks to find
    - Bug was that multiply required 2^{-47} bit to be reset and the reset was being decoded spuriously on occasions

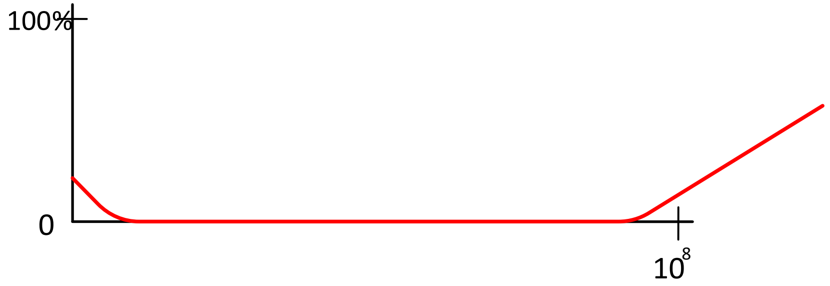
Analog scanners used relays to sequentially route the input to the A/D converter. These were failing in much larger numbers than had been expected. Scanning was automatic and results passed into store via DMA.

- Reed relays were rated at 10^8 operations. At one per second these should have lasted over 3 years or 30 years for mercury wetted relays. Some sites would not allow mercury on site!
- I built a rig to test lifetime at an accelerated rate of 100 per second for up to 24 relays
- The results were most unexpected

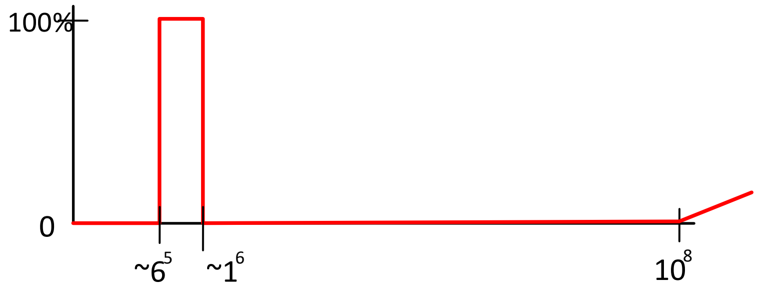
Relay test results

The test bed used single bit outputs to operate the relays and single bit inputs to test the contact state. A few milliamps were passed through the contact.

Expected a typical “bathtub” curve



What I saw (and repeated) was



This applied for BOTH dry and mercury wetted relays although the failure point was slightly different

Relays were tested by manufacturers at beginning of life and at samples throughout their life but this failure mode had not been found before.

The solution was to “run-in” all relay boards for a million operations – no more failures were reported

Relay reed manufacturers asked me to test new design sample and these were OK. I never did get an explanation as to how it was possible!

References

1. A new microminiature digital computer (C.S Evans & P Woltenholm – date unknown)
   - ○ https://www.sciencedirect.com/science/article/pii/S147466701768995X?via%3Dihub