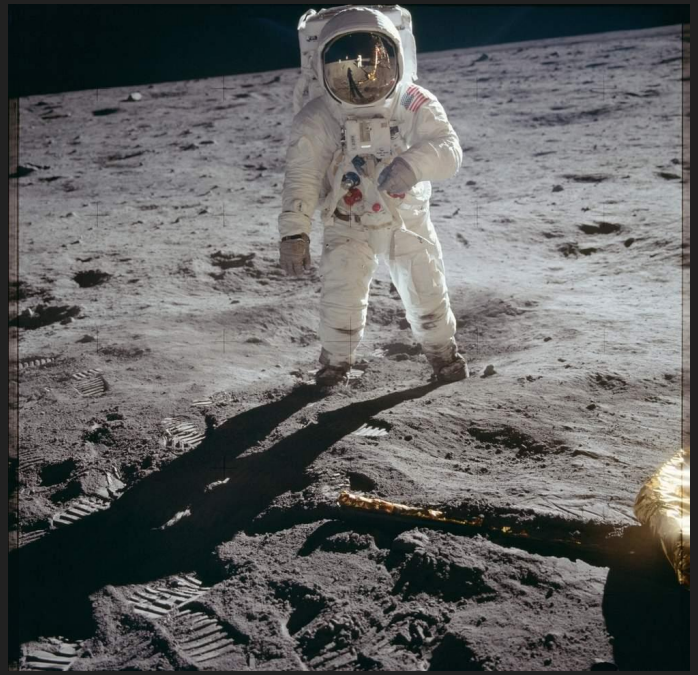
MIPS vs. AGC

The Advances of Computer Architecture

Annie Kroo, March Saper, Paige Pfenninger

The Apollo Guidance Computer

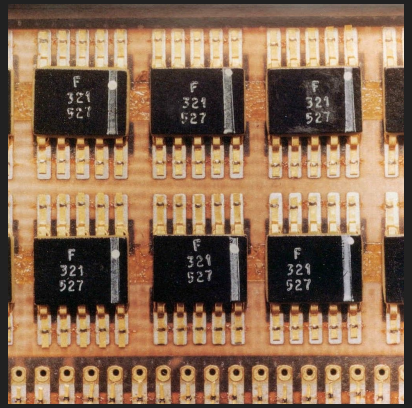
Context



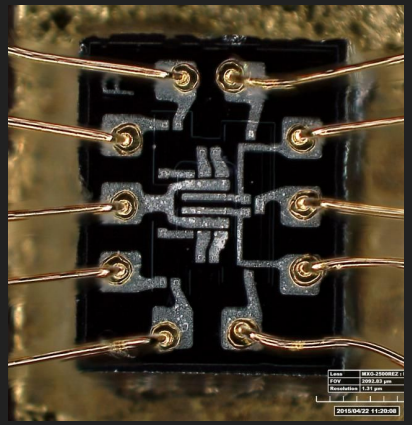
- 1969 Moon Landing
- Two versions
  - Block 1: Used for unmanned missions
    - 4100 single three-input Nor gate
  - Block 2: Used to land on the moon
    - 2800 dual three-input nor gates
- Some of the first computers to use ICs
- Developed by MIT
  - Margaret Hamilton
  - “Rushed Job”
- Very few established standards - new

The Apollo Guidance Computer

The Pieces



Flatpack ICs used in ACG



Microscopic view of Apollo NOR gate in chip

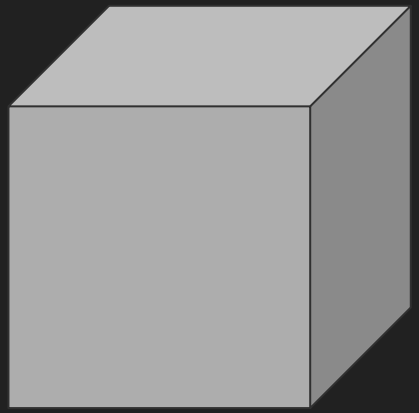
DSKY Interface

One’s vs. Two’s Complement

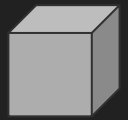
- Up until 1969 2’s Complement not standard
- The AGC used 1’s Complement
- One’s Complement has two zeros
  - Positive Zero (0000)
  - Negative Zero (1111)
- Makes Arithmetic difficult

Decimal	Two’s C.	One’s C.
-4	100	XXX
-3	101	100
-2	110	101
-1	111	110
0	000	111 & 000
1	001	001
2	010	010
3	011	011

Memory



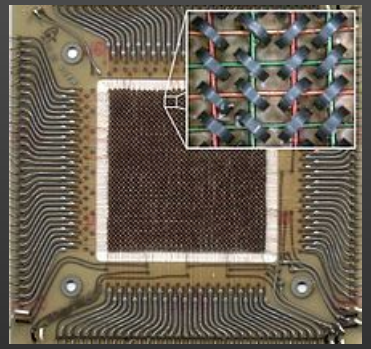
256 Cubic Feet



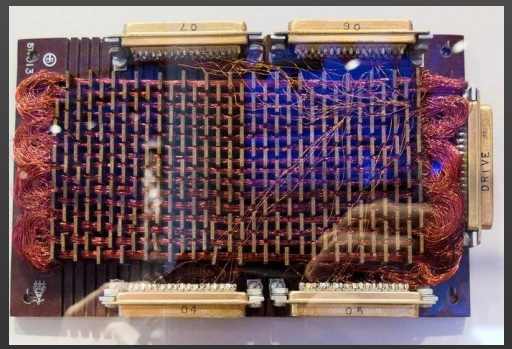
14 Cubic Feet

←—— 9^-10 Cubic Feet

Magnetic Core Memory

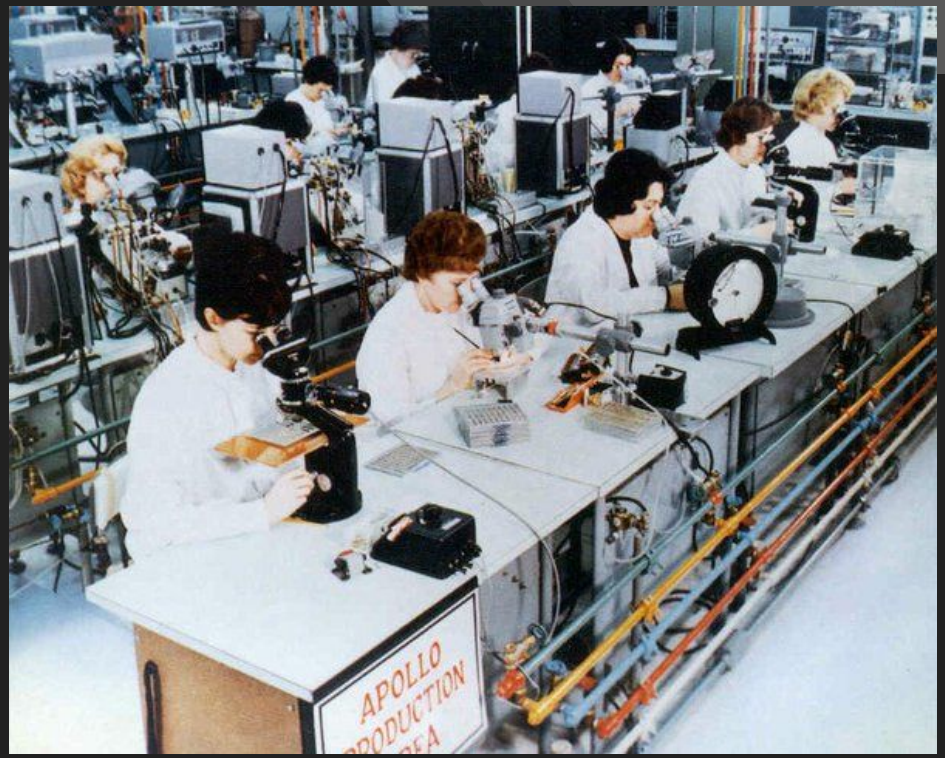


Core Rope Memory



"Software as hardware"

Little Old Lady Memory



Registers

32 Registers in MIPS

8 Registers in the AGC

- The eight registers all had very specific purposes and could only be used for one thing
- The A register (the accumulator) was used as both an input and an output to all mathematical operations
- The registers were stored in the first eight words of the erasable memory
  - There was no separate register file

Memory Addressing

- 16 bits were needed to fully address all memory locations
- The memory address was only 12 bits long

AGC

Need to use a combination of:

- 12 bit memory address
- 5 bit fixed memory bank
- 3 bit erasable memory bank
- 1 bit super bank

MIPS

Get the entire memory address from one of the registers

CHART 3

ADDRESS SELECTION LOGIC

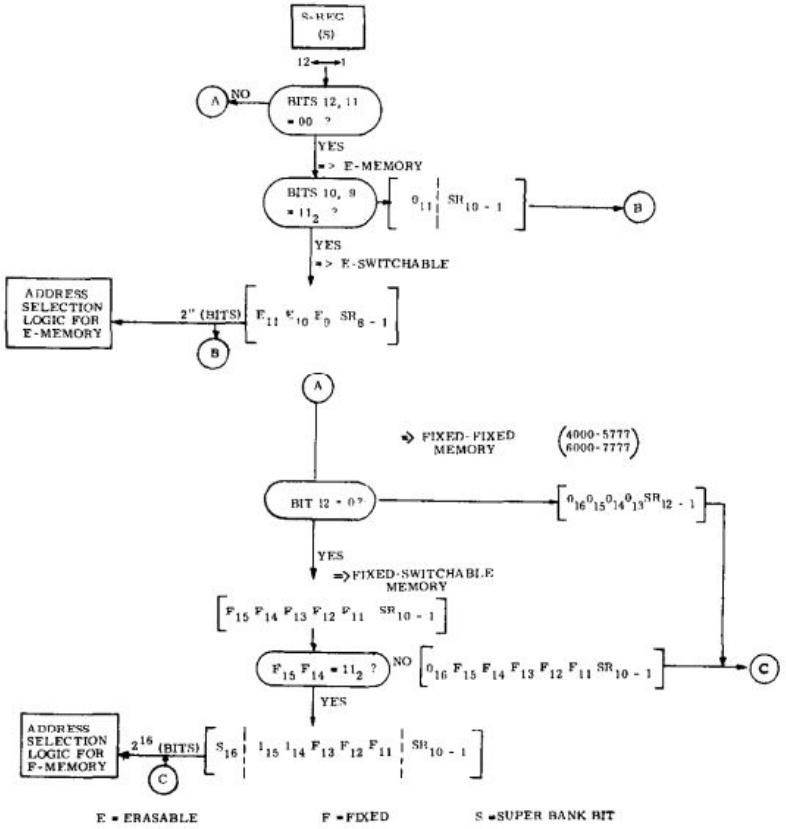
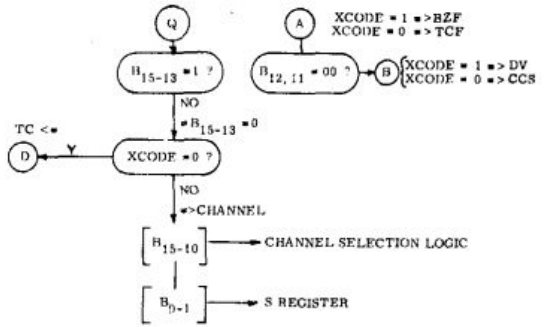
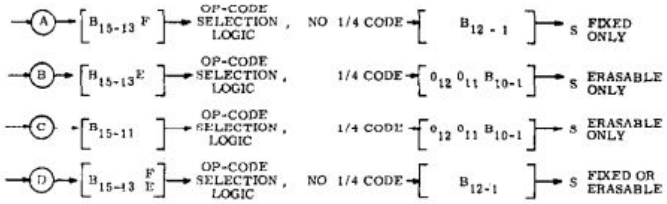


CHART 3 continued



F = ERASABLE F = FIXED B = B-REG S = S-REG



AGC Instructions

- Instruction format: 3-bit op code and 12-bit address
- Block 1 had 11 instructions
  - And with these 11 instructions we made it into space!
- Many instructions have analogs to what we saw with MIPS
  - Implementations of addition, subtraction, jump etc.
- But there were a few funky ones too

INDEX instruction

- Adds the data retrieved at the address specified by the instruction to the following instruction

Example Program *(Instr, Addr))*:

INDEX K
MASK L

Memory

K	Some Constant
...	...

Actual Execution

INDEX K
MASK L + Some Constant

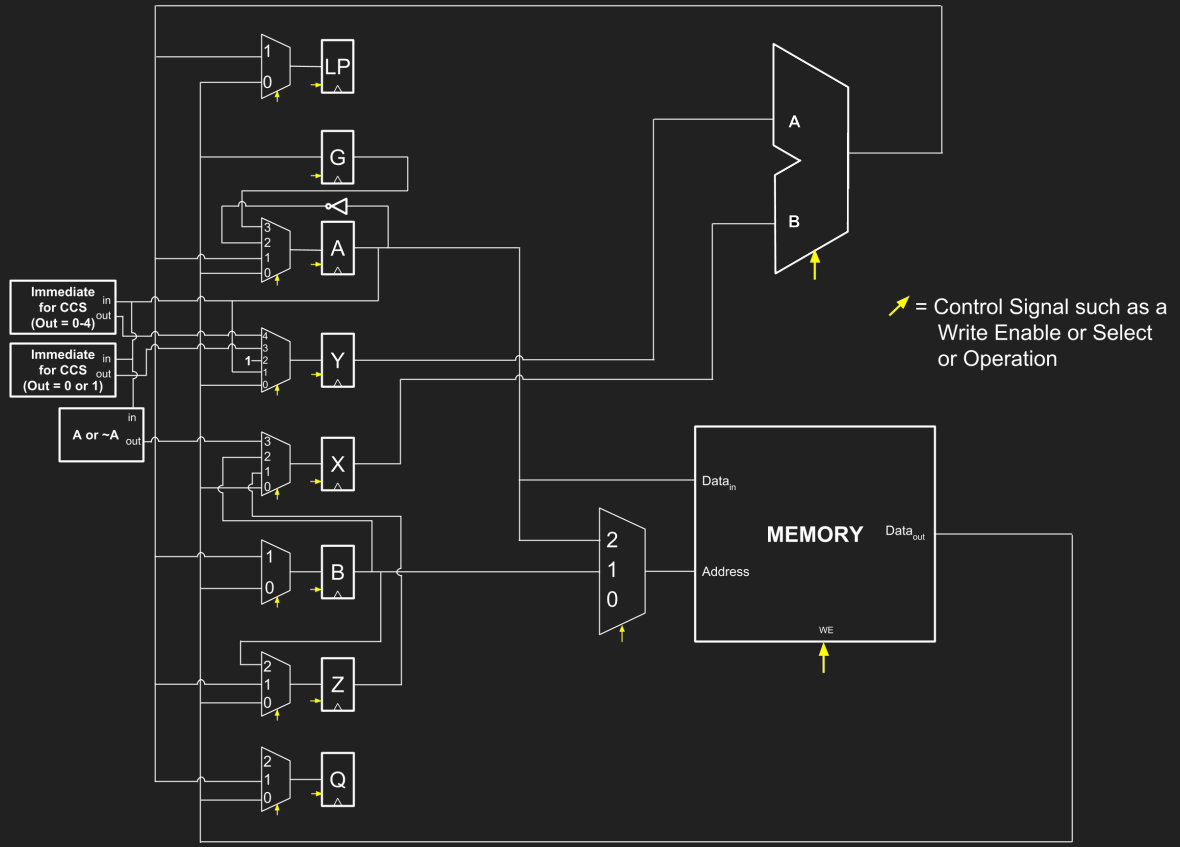
Three potential results:

- Indexing the data if no 12-bit overflow
- Changing the instruction type if 12 to 14-bit overflow
- “Extending” an instruction if 15-bit overflow. Works if “some constant” = 100111111111111
  - Additional EXTEND instruction actually implements INDEX d’3071

Fun Instruction: CCS

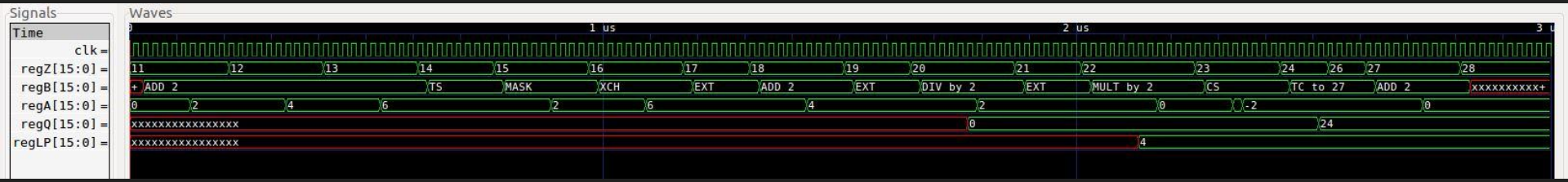
- Used for conditional jumps and loop control
- Compares a value in memory to a known constant and increments the program counter by 1, 2, 3, or 4 depending on the result
- If there was no way for an instance of CCS to result in a jump of 1 or 2, this created “holes” in the memory.
    - A “CCS-Holes” task force was assigned to find these holes and use that memory space for constants

Our Simulation



Block Diagram of Our Implementation

It Works!



EXPECTED RESULT					ACTUAL RESULT					
regZ	regB	regA	regLP	regQ	regZ	regB	regA	regLP	regQ	PASS:
11	AD	2	X	X	11	AD	2	X	X	Y
12	AD	4	X	X	12	AD	4	X	X	Y
13	AD	6	X	X	13	AD	6	X	X	Y
14	TS	6	X	X	14	TS	6	X	X	Y
15	MSK	2	X	X	15	MSK	2	X	X	Y
16	XCH	6	X	X	16	XCH	6	X	X	Y
17	EXT	6	X	X	17	EXT	6	X	X	Y
18	SUB	4	X	X	18	SUB	4	X	X	Y
19	EXT	4	X	X	19	EXT	4	X	X	Y
20	DIV	2	X	0	20	DIV	2	X	0	Y
21	EXT	2	X	0	21	EXT	2	X	0	Y
22	MP	0	4	0	22	MP	0	4	0	Y
23	CS	-2	4	0	23	CS	-2	4	0	Y
24	TC	-2	4	24	24	TC	-2	4	24	Y
27	AD	0	4	24	27	AD	0	4	24	Y

Overall: **PASS**