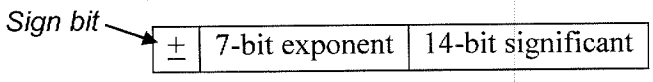
Konrad Zuse's Z1

Architecture:

- 64 word x 22 bit memory
- Two registers: R1 and R2
- First load loaded R1; subsequent loads loaded R2
- Memory stores had no operand – always stored R1
- After a store, next load was to R1
- Instructions included: addition, subtraction, multiplication, and division
- All arithmetic operations combined R1 and R2, placed the result in R1, and cleared R2.

Zuse's Floating-Point Representation



Zuse's Z1 Instruction Set

Instruction/Description	Opcode	Cycles
Lu – read keyboard	01 110000	9 to 41*
Ld – display result	01 111000	9 to 41*
Pr z – load from address z	11 $z_6z_5z_4z_3z_2z_1$	1
Ps z – store to address z	10 $z_6z_5z_4z_3z_2z_1$	0 or 1
Lm – multiplication	01 001000	16
Li – division	01 010000	18
Ls1 – addition	01 100000	3
Ls2 – subtraction	01 101000	4 or 5

* - depending on exponent

Manchester "Baby"

Architecture:

- Memory: 32-bit by 32 word
- Program flow controlled with control register, CI (program counter): incremented by one immediately before executing instruction
- 32-bit accumulator
- Arithmetic unit could only subtract and negate

SSEM Instruction Layout

Bit position:	0	1	2	3	4	5 .. 12	13	14	15	16 .. 31
Function:	Memory address					Unused	Function			Unused

SSEM Instruction Set

Binary function	Modern Mnemonic	Instruction/Description
000	JMP	Load CI with address stored in location pointed to by instruction (indirect jump)
100	JRP	Retrieve offset stored in location pointed to by instruction and add it to CI (relative jump)
010	LDN	Copy to accumulator the negated contents of the memory location pointed to by instruction
110	STO	Copy contents of accumulator to memory location pointed to by instruction
001	SUB	Subtract from accumulator the contents of the memory location pointed to by the instruction
101	-	Not defined (If used, SSEM performed SUB function)
011	CMP	Skip the next instruction if the content of the accumulator is negative
111	STOP	Halt the machine

IAS Machine

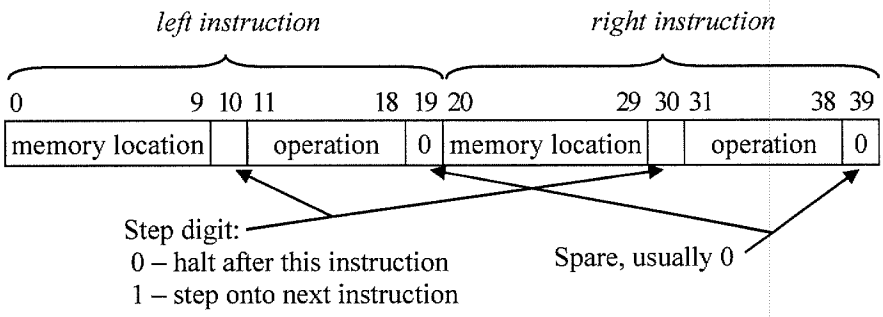
Two 40-bit registers in the arithmetic unit:

- Accumulator (AC) - held result of an arithmetic operation, upper half of a multiplication result, remainder after a division, multiplicand for a multiplication, or dividend for a division.
- Multiplier Quotient (MQ) - held lower half of a multiplication result, quotient after a division, or multiplier for a multiplication.

Five registers in control unit:

- Program Counter (PC) – 10-bit register contained the address of the next 40-bit instruction pair to fetch from memory
- Memory Buffer Register (MBR) – 40-bit register contained the word (datum or instruction) to be written to or most recently read from memory
- Memory Address Register (MAR) – 10-bit register contained the address for a transaction to or from memory (datum or instruction)
- Instruction Buffer Register (IBR) – 40-bit register contained the most recently fetched instruction pair
- Instruction Register (IR) – 20-bit register held the instruction currently being executed

Princeton IAS Machine Instruction Layout



Princeton IAS Machine Instruction Set

Opcode (binary)	Operation description
11100101	Load AC with value from address in memory
11100100	Add value from address in memory to the AC
11101101	Load AC with negative of value from address in memory
11101100	Subtract value from address in memory from the AC
11110101	Load AC with absolute value of number from memory
11110100	Add absolute value of number from memory to the AC
11111101	Load AC with negative absolute value of number from memory
11111100	Subtract absolute value of number from memory from the AC
11100000	Multiply MQ with value from address in memory; put upper half of result in AC and lower half of result in MQ
11100010	Multiply MQ with value from address in memory; put only upper half of result in AC to round off lower 40 bits
11111000	Divide AC by value from address in memory; leave remainder in AC and put quotient in MQ
11100110	Load MQ with value from address in memory
11010100	Store AC to address in memory
11010101	Store AC to address in memory then clear AC
11010000	Unconditional jump to left instruction of destination address (step digit = 0)
11010000	Unconditional jump to right instruction of destination address (step digit = 1)
11010000	If value in AC is non-negative, jump to left instruction of destination address (step digit = 0)
11011000	If value in AC is non-negative, jump to right instruction of destination address (step digit = 1)
10100000	Shift AC right one position with LSB shifting into MQ
10100010	Shift AC right one position discarding LSB
10101000	Shift AC right one position discarding MSB
10100100	Transfer MQ to AC
N/A*	Replace address portion of left instruction at address in memory with left most 10 bits in AC
N/A*	Replace address portion of right instruction at address in memory with left most 10 bits in AC

* These instructions were part of the original IAS proposal, but do not appear as implemented in the final progress report