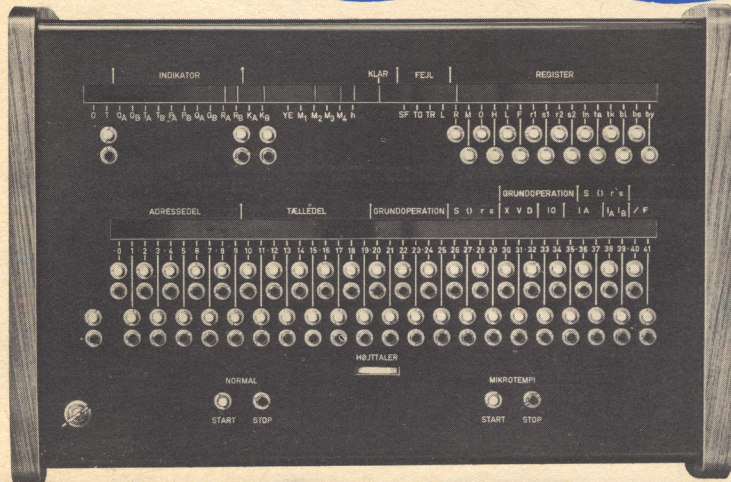


disa

computer
system

dec

compiler
assembler
subroutines



cuts cost *of* labor

previously

2 weeks of calculation
6 days of calculation
1 day of calculation

now

14 minutes
2 minutes
1 second

disa
dec

*saves money for its customers
- in technical design and
in pure scientific analysis.*





The customers of her company change their minds. But with this computer her company follows them up from modification to modification: Inquiries are telexed in. If it is a hot proposal, then: details are spelled out, programs are chosen and put on the computer. Half an hour later computing results are being printed out and become basis for the proposal being telexed back next or same day. And, the client has hardly had time to contact any competitor.

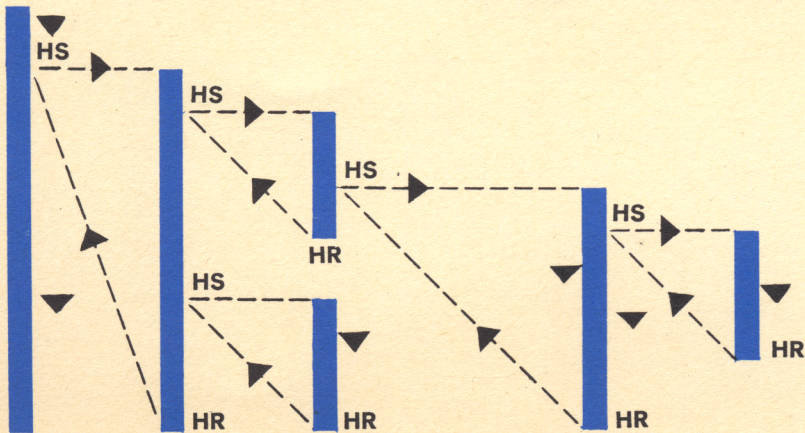


This research center maintains its own courses of instruction in the programming of the machine. After 3-4 days of instruction their people write their own programs in compiler language. Presently, they have 28 persons who program in compiler language and 5 persons who program in machine language and, they possess already a backlog of around 60 programs and 10-20 subroutines.

special features of programming

disa
dec

Main program



- Each and every instruction can be made conditional by specifying an N or L modification.
- The jump instructions HS and HR make it easy to use subroutines:
This is accomplished in the machine by an automatic storing and resetting of jump-addresses. Every jump to a subroutine shall be an HS-instruction, every return jump shall be an HR instruction. (See illustrations).
- Every operand contains two marking bits. The N and L modifications enable us to accept or reject data based on the marking.
- The multiplications MK and ML of fixed point numbers are accumulating:
During MK the initial value of Result Register is being added to the high order part of the product of the cell and Multiplier Register.
During ML the initial value of Result Register is being added to the low order part of the product.

disadec's complete vocabulary of instructions

Arithmetic

AR(F)	Add cell to Result Register
AN(F)	Add absolute value to Result Register
AC	Add Result Register to cell
AB	Add Boolean to Result Register
SR(F)	Subtract cell from Result Register
SN(F)	Subtract absolute value from Result Register
SC	Subtract Result Register from cell
MK(F)	Multiply short
ML	Multiply long
MT	Multiply with sign of cell
MB	Multiply Boolean
DK(F)	Divide short
DL	Divide long

Shift and Number Conversion

NK	Normalize short
NL	— long
TK	Short open shift
TL	Long — —
CK	Short round —
CL	Long — —
NK F 0	Number in Result Register from fixed to floating point
TK F 10	Number in Result Register from floating to fixed point

Storing in cell

GR(F)	Store Result Register
GM	Store Multiplier Register
GA	Store unmodified address part of Result Register in cell
GT	Store address decrement part of Result Register in cell
GP	Store index register in address part of cell
GS	Store subroutine register in address part of cell
GI	Store indicator register in address part of cell
GK	Store (by-register) in address part of cell, and store track register in address decrement part of cell

PA	Store address decrement in address part of cell
PT	Store address decrement in address decrement part of cell

Transfer to register

PM	Transfer cell to Multiplier Register
PP	Transfer resulting address to index register
PS	Transfer resulting address to subroutine register
PI	Transfer modified address to indicator register with reflected address decrement as mask

Jumping and Execution

HV	Jump to left half instruction
HH	Jump to right half instruction
HS	Jump to subroutine
HK	Jump to subroutine if drum is free, otherwise continue
HR	Jump return from subroutine
UD	Execute one instruction and return

Conditional instructions ("skip" means: skip unto and including next right half instruction)

BS	If modified address $\leq$ address decrement, then skip.
BT	If resulting address $\leq$ address decrement, then skip.
CA	If not coincidence between resulting address and unmodified address part of Result Register, then skip.
CN	If coincidence between resulting address and unmodified address part of Result Register, then skip.
CM	If coincidence between cell and Result Register using Multiplier Register as mask, then skip.

Address modifying instructions

IS	Apply resulting address as s-value in immediately following s-modified instruction
----	--

IT	Apply resulting address as address decrement in immediately following instruction
----	---

Drum

VK	Select drum track by transferring address to tk-register
LK	Read from selected drum track to core memory
SK	Write on selected drum track from core memory

Input - Output

VY	Select input output equipment by transferring modified address to by-register with reflected address decrement part as mask
LY	Read from selected input unit to bits no. 3-9 of Result Register and of cell
SY	Write bits no. 3-9 of resulting address on selected output units

Auxiliary instructions

QQ	Dummy operation, but execute specified modifications
ZQ	Stop after execution of specified modifications
XR	Exchange Multiplier Register and Result Register

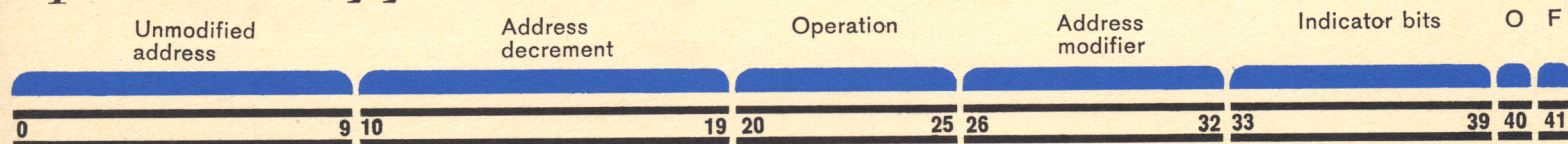
Modifications of instructions

S	Clear Result Register before basic operation
X	Execute basic operation, and exchange Multiplier Register and Result Register
V	Skip next full cell
D	Use resulting address as operand
F	Execute operation in floating point
I	Transfer to indicators
M	Store marking bits in cell

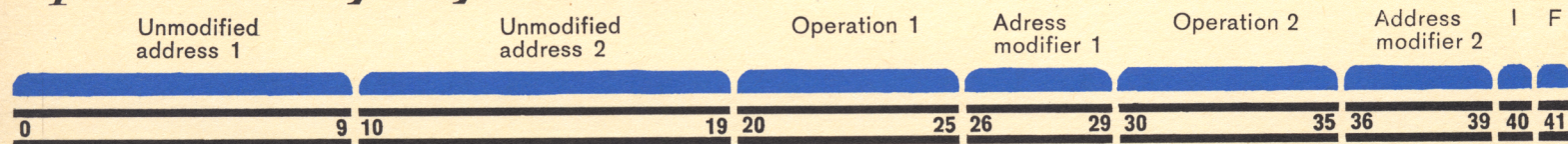
Conditional modifications

L	Execute instruction only if specified bit is 1
N	Execute instruction only if specified bit is 0

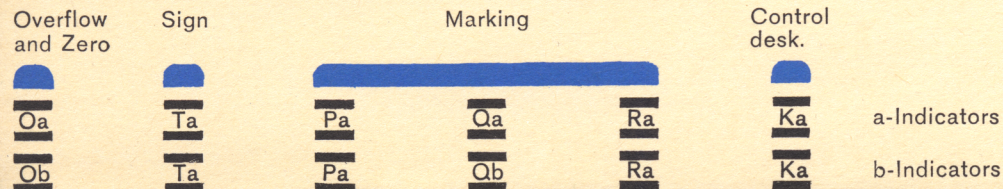
representation of full instructions



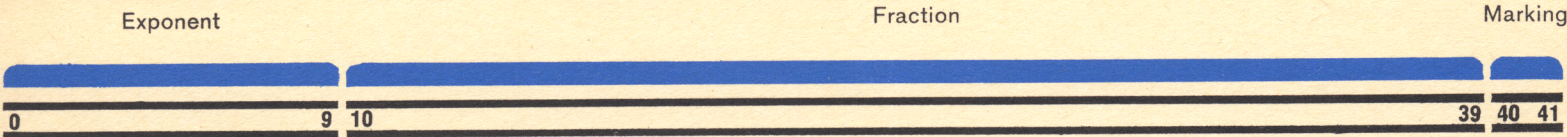
representation of half instructions



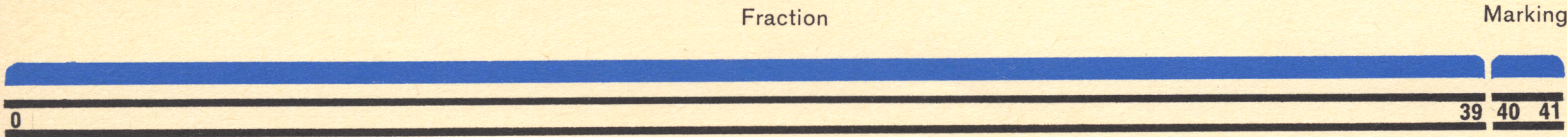
representation of indicators



representation of floating point numbers



representation of fixed point numbers



Fixed Point:	Add and Subtract	22 μ Sec.
	Multiply	155 μ Sec.
	Divide	240 μ Sec.
Floating Point:	Add and Subtract	66 μ Sec. + 2.2 $\times$ (exp. + norm.) μ Sec.
	Multiply	140 μ Sec.
	Divide	190 μ Sec.

specifications

Fully transistorized, binary, digital computer.
Parallel, synchronous operation.

Software:

Assembler SLIP for symbolic addressing.
Algol compiler
Established **library of subroutines**.

Words of 40 bits + 2 marking bits.
A word equals 1 full instruction or 2 half instructions or 1 operand.

The 2 **half** instructions provide for increased computing power per word being fetched.

Negative **numbers** by 2-complement.
Decimal numbers and alphanumeric symbols can be processed as well.

Any instruction can be conditional (not only JUMP instructions).

Vocabulary of 57 instructions, each with one address.

Complete indirect and relative addressing allows all **programs** to be **fully relocatable**.

1 fast register + entire memory available as index registers.

All full instructions contain address decrement for automatic indexing.

Relative addressing to any higher level of subroutines. Automatic return through arbitrary levels of subroutines.

The marking bits make it easy to use grouping of data.

Hardware:

Fast memory: 1 K ferrite core memory 9 μ Sec. cycle time.

Large capacity memory: Drum with 12800 words on 320 channels of 40 words.

95 % of drum transfer time available for computing.

0,45 M c/s clock
Readily exchangeable microprogram unit.

1 kW, 380 V, 3 phases.
Dimensions: 180×160×40 cm (70×62×16 inches)
Weight: 500 kg (1100lbs)

Auxiliary equipment

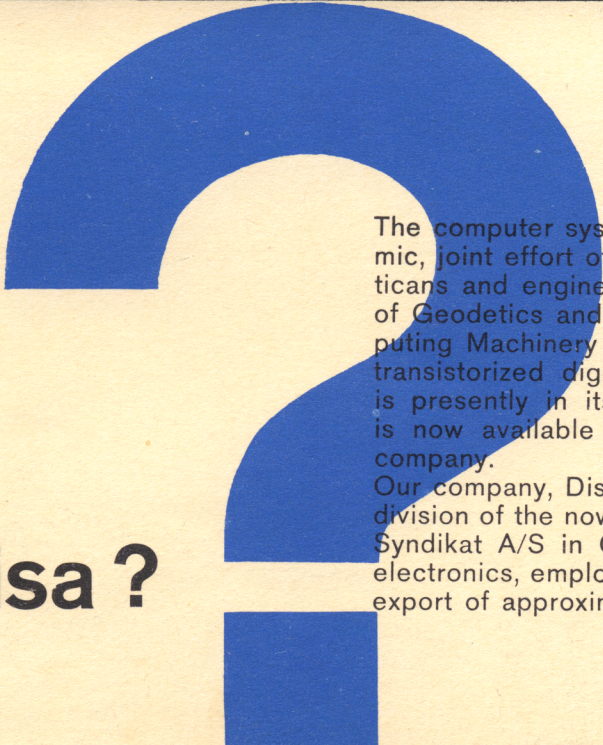
On line: 500 characters/sec. paper tape reader (Facit ETR 500)
Input-Output typewriter
150 characters/sec. paper tape punch (Facit PE 1500)
Off line: Flexowriter

The system can be expanded with

Lineprinter: 1000 lines/minute type ANelex

Magnetic Tape unit: Min. 1 sec. access
Max. 3 sec. access
200 characters/inch. 200 inc./sec. (Facit ECM 64)

Punch Card Reader: 42000 cards/hour
80 holes and 27 pencil marks per card.



what is Disa ?

The computer system Disadec is the result of a dynamic, joint effort of two Institutes: A team of mathematicians and engineers from The Royal Danish Institute of Geodetics and from The Danish Institute for Computing Machinery designed and built 1960-61 the fully transistorized digital computer GIER-DISADEC. This is presently in its second year of operation, and it is now available to the general public through our company.

Our company, Disa Elektronik, was founded 1946 as a division of the now some sixty years old Dansk Industri Syndikat A/S in Copenhagen. We produce precision electronics, employ some 270 people, and we enjoy an export of approximately 80 % of our business volume.



Disa Elektronik A/s

17 Herlev Hovedgade, Herlev, Denmark
Telegram: Disatron Copenhagen.
Telex 5849 Syndikat, Copenhagen
Telephone: Copenhagen 94 52 11.