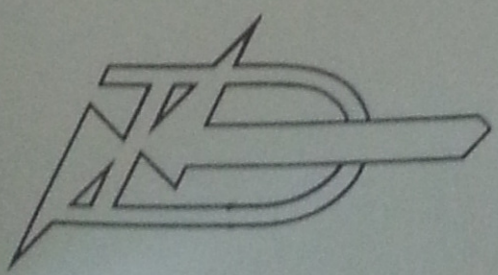
G. BENTSEN

HARDWARE MANUAL
VOLUME II
FLOW DIAGRAMS

NORD-1
COMPUTER SYSTEMS

Hardware Manual
Volume II
Flow Diagrams



A/S NORSK DATA-ELEKTRONIKK

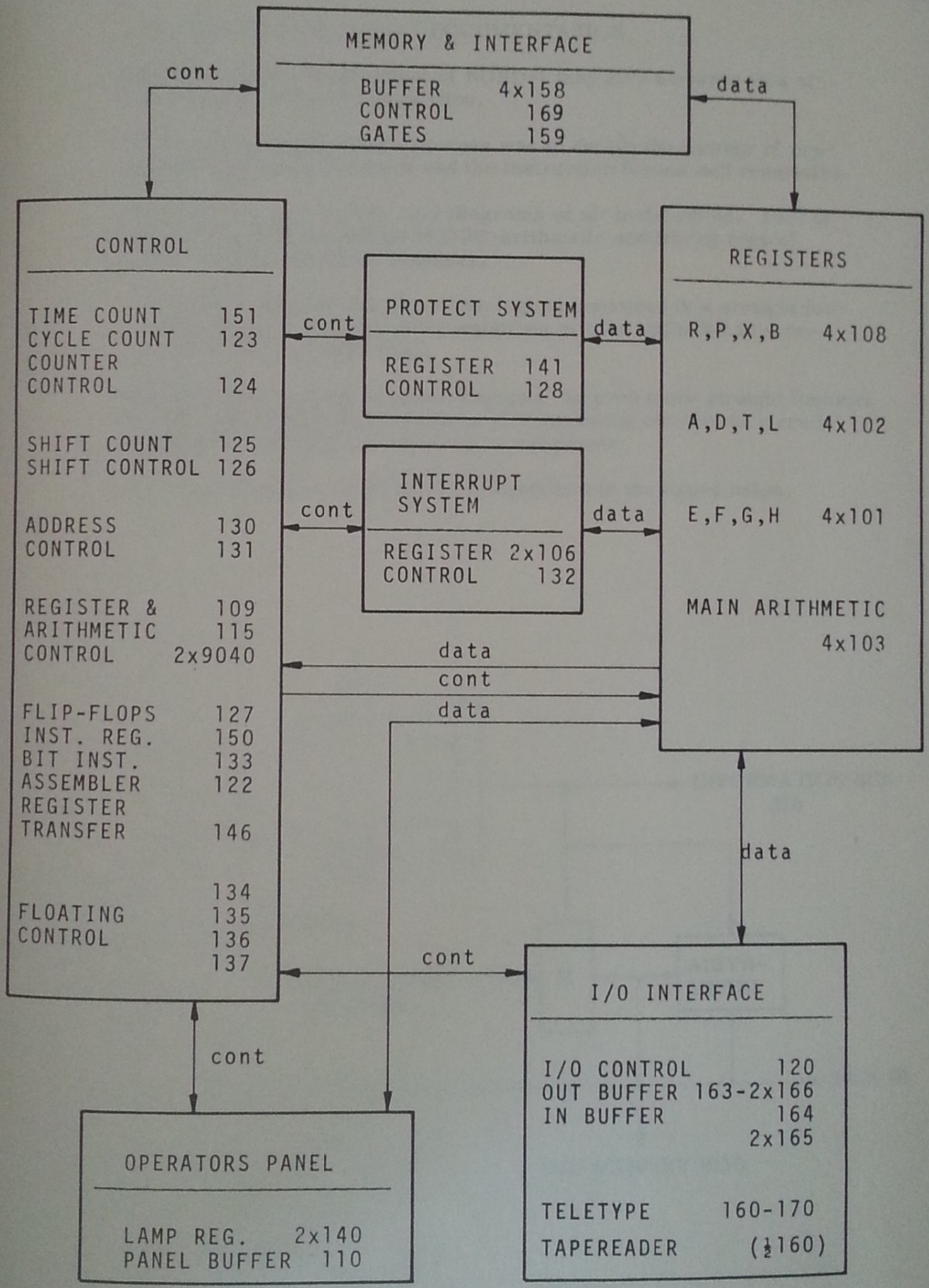
TABLE OF CONTENTS

1 Introduction

2 Flow Diagrams

3 Signal Definitions

A/S NORSK DATA-ELEKTRONIKK	Title: NORD - 1	Drawing no.: Page 1



DRAWN BY EML	Remarks	Replacement for	Date
APPROVED BY			
DATE 31.10.69		Replaced by	Date

INTRODUCTION TO NORD-1 DOCUMENTATION

Summarizing the development of NORD-1 may give a coarse idea of the structure of the documentation.

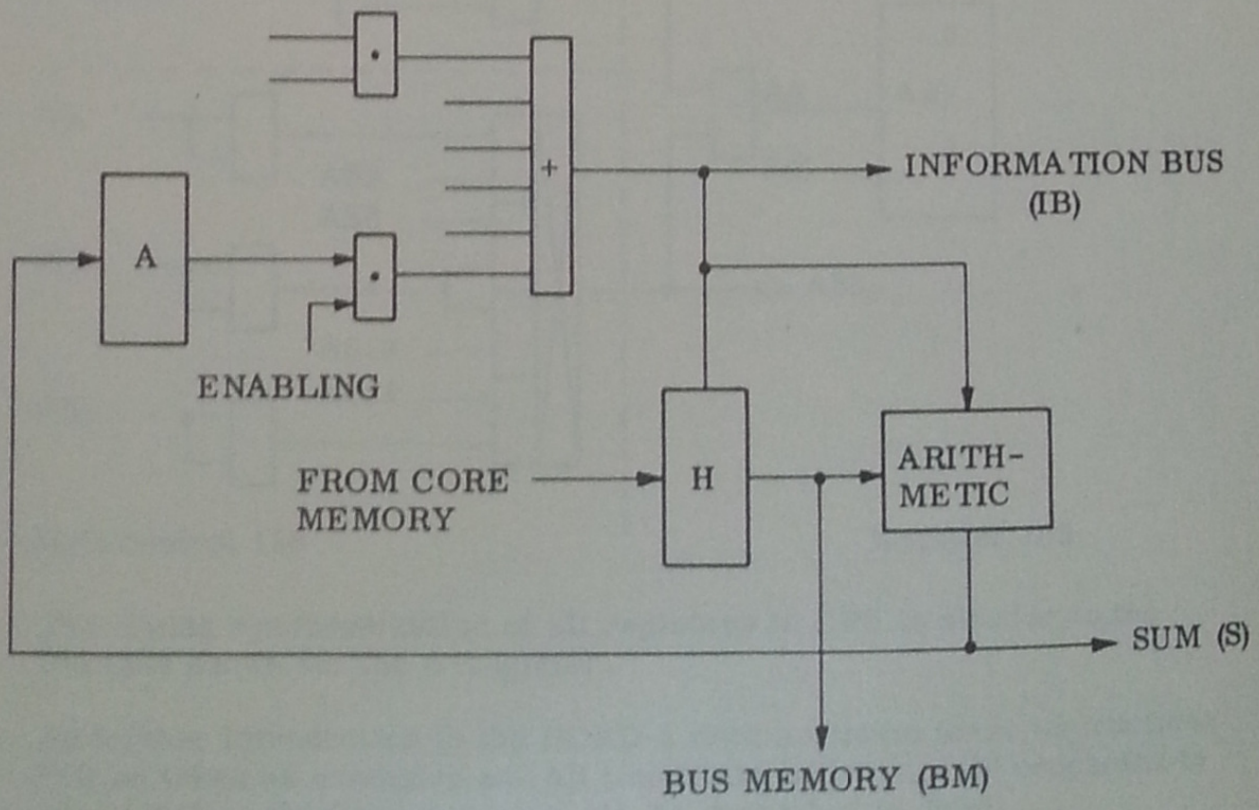
The first step in the design process was to decide the number of programmer oriented registers and the instruction format and repertoire.

The next step was to draw flow diagrams of all instructions. Part of this work is also the design of CPU-arithmetic and timing control (Time Counter and Cycle Counter).

To translate the flow diagrams into logical equations is a straight forward mechanical work, and the equations may be regarded as a rewriting of the flow diagrams.

The last step, drawing of logic diagrams, is even more straight forward. The problem at this stage consists of distributing circuits on circuit boards and definition and labelling of subsignals.

The basic operation of NORD-1 is illustrated in the figure below.

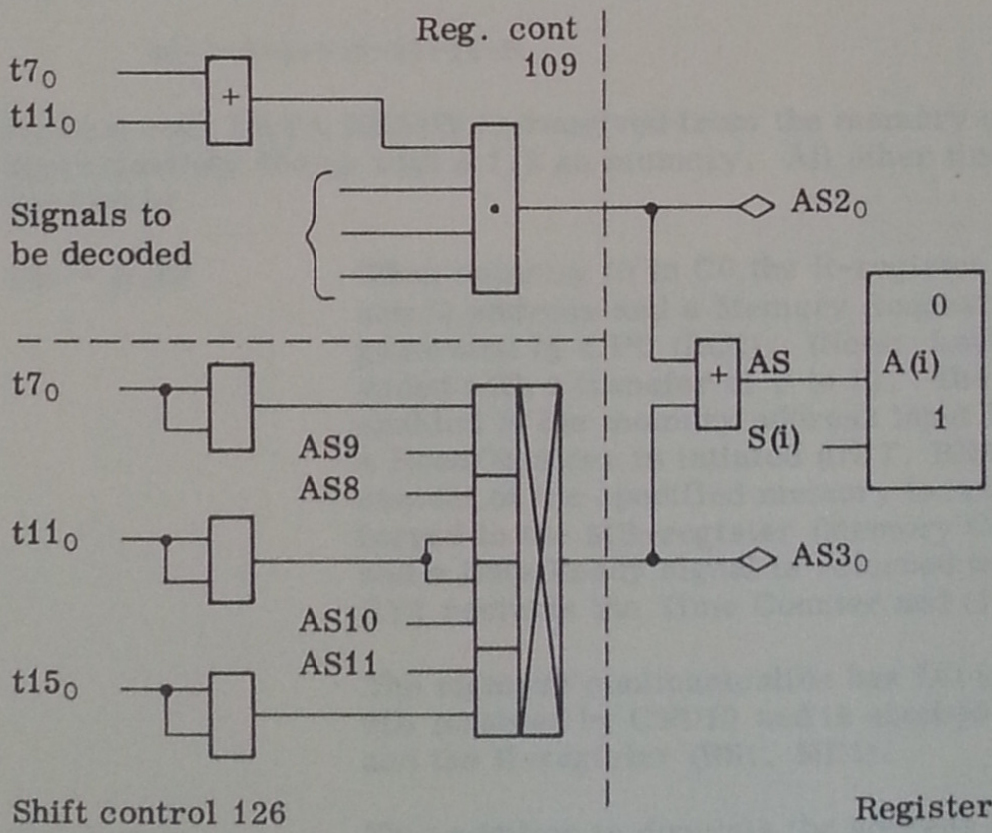


One register is enabled at a time and presented at the IB terminals, and as input to the arithmetic. Data from memory goes via the H-register and the BM input to the arithmetic. For inter-register operations one of the operands (source) is first transferred to the H-register before IB is switched to the other operand. The output of the arithmetic is then routed back to all registers where it may be strobed into register flip-flops.

The purpose of the control logic therefore is to generate the appropriate sequence of register enabling signals and register strobe pulses and to control the arithmetic to give the desired function, all as a result of the instruction decoded.

The operation of the logic is synchronous, i.e. controlled by a common timing source (Time Counter and Cycle Counter). To avoid timing hazards the timing signals, after being decoded from the time counter flip-flops, always passes three stages of logic circuits before reaching the register flip-flops.

Example: Generation of A-register set pulse.



The timing synchronization of all registers in CPU is similar to the example shown for the A-register.

As further introduction to the NORD-1 documentation three instructions will be taken as examples and all important active signals connected to each block in the flow diagrams will be described in detail.

The example instructions are:

1. add 120
2. skp dx gre zro
3. shad rot 5

The reader is advised to find the mentioned signals in the logic diagrams, to familiarize himself with the documentation.

add 120

Relevant flow diagrams: Pages 1 and 12

The instruction add is a two-memory cycle instruction requiring two 16 bit words to be read from core memory, first the instruction itself (C0) and then the data word (C2).

The Time Counter control signals BACK and NEXT are both 0 in both cycles, and the Time Counter therefore produces the short sequence

t0-1-2-3-8-9-10-11-0

t0 lasts until DATA READY is received from the memory control and is approximately 800 ns with a 1.5 μs memory. All other time intervals are 100 ns.

(R) → H,IR

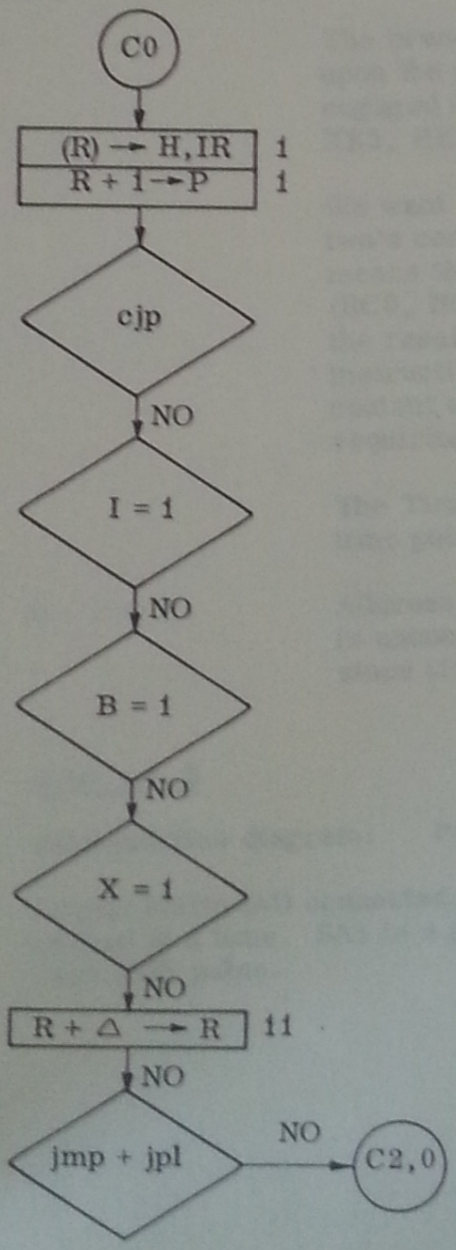
When entering t0 in C0 the R-register contains the actual address and a Memory Request signal is generated by CPU (RQ1). (Note: Last instruction ended with a transfer of P to R). The address is enabled to the memory address input MR (158), and a Read/Restore is intiated (INIT, RRA 169). The content of the specified memory location is transferred to the MB-register (Memory Control Module) and a Data Ready signal is returned to CPU (RY1). RY1 restarts the Time Counter and t1 is entered.

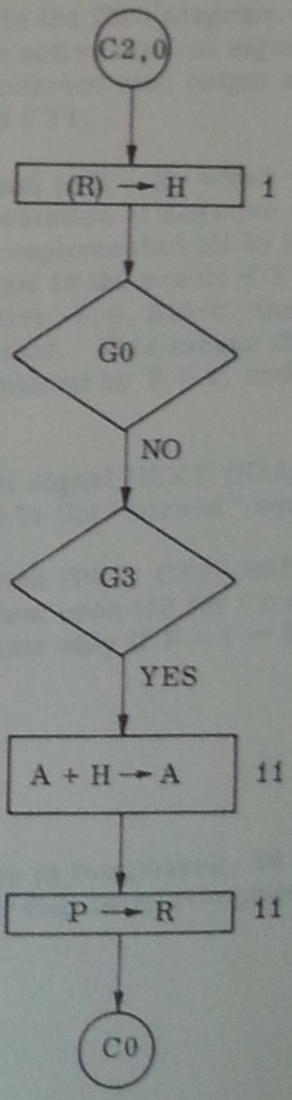
The memory communication bus MJ is now equal to MB (enabled by CPUE) and is strobed into IR (IS1) and the H-register (HS1, ME1).

R + 1 → P

This addition is done via the address arithmetic. First P is set to all ones (PS3) and then the appropriate zeroes are transferred to P via the flip-flop clear terminal (PS4). RE1 enables the R-register as input to the arithmetic and CY2 generates the 1 to be added as a carry input to the least significant stage of the adder.

R + 120 → R	Address arithmetic addition (RE1, HE2, SX1, RS1). R is equal to the address of current instruction. 120 is the displacement of the instruction with sign extension, i.e. bits 8 through 15 equal to bit 7. 120 is transferred via the H-register and SX1 controls the sign extension mechanism. The flip-flop D-input is used since the input to R is a function of R itself.
(R) → H	Similar to (R) → H,IR
A + H → A	Input to the arithmetic is enabled by HE1 and AE1, which makes BM=H and IB=A. The output of the arithmetic is the sum of BM and IB if no other control signal is active. (PLUS=$XR_0ID_0BD_0$). The sum is strobed into A at t11 by AS2.
P → R	Transfer via address arithmetic (PE1, RS1).





skp dx gre zro

Relevant flow diagram: Page 5

This instruction is a one cycle instruction (C0). The Time Counter control signal NEXT depends upon resulting skip condition and selects one of the following two time pulse sequences.

skip not effective: t0-1-2-3-8-9-10-11-0
skip effective: t0-1-2-3-8-9-10-11-12-13-14-15-0

(R) → H,IR — same as example 1

R + 1 → P — same as example 1

S.R. → H — In the instruction above the source register code is 0, which means that no register is enabled and therefore IB=0. This zero is transferred to the H-register (IE1, HS1).

P → R — same as example 1

The branching conditions in the flow diagram depends upon the sum output. The active control signals engaged in producing the correct sum output are XE3, HE1, BC0, BC8 and CY1.

We want the sum to be equal to X - H, which with two's complement representation of negative numbers means that H should be complemented bit by bit (BC0, BC8) and a one added to the result (CY1). If the resulting sum is positive, i.e. S15=0, the next instruction should be skipped. This means that the content of R should be replaced by P + 1, and operation requiring additional time.

The Time Counter control signal NEXT (NX1) adds time pulses t12-13-14-15 to the "normal" sequence.

P + 1 → R — Address arithmetic addition (PE1, CY2, RS1). RS1 is unconditionally dependent upon t15 for C0:skp since t15 in this case exists only if P + 1 → R is wanted.

shad rot 5

Relevant flow diagram: Page 7

Double shifts (AD connected, 32 bits) are done in two phases, 16 bits shifted at a time. SA1 is a phase control flip-flop being complemented at each shift pulse.

SA1 = 0: shift the D-register

SA1 = 1: shift the A-register

No shift direction specified means left shift.

(R) → H,IR	Same as example 1
R + 1 → P	same as example 1
0 → SA1, SC	Reset Shift Counter (SC) and the phase control flip-flop (SA1).
5 → SC	5, the number of places to be shifted, is transferred from the H-register to the shift counter via the main arithmetic (required by floating point instructions). (HE1, BD, SS8).
SC - 1 → SC	If SA1 = 1 the shift counter is decremented.
SA_0 → SA	Complement the phase control flip-flop.
shift D	If SA1 = 0 shift the D-register one place to the left. (DE1, IL, ID, DS2, DX1).
	IL enables left shift of IB. ID makes the output of the arithmetic equal to the IB-input and DX1 selects A15 as input to D0. The discarded bit (D15) is transferred to the link flip-flop M.
shift A	If SA1 = 1 shift A-register one place to the left (AE1, IL, ID, AS3, DX1). In this case DX1 selects M as input to A0, and the discarded bit (A15), which has already been transferred to D0, replaces the previous content of M.
	If the content of the shift counter is not zero go back to SC - 1 → SC. This is controlled by the Time Counter control signal BACK (BK1), which repeats pulses t8-9-10-11 the required number of times, ten in our case, five for each phase.
P → R	same as example 1

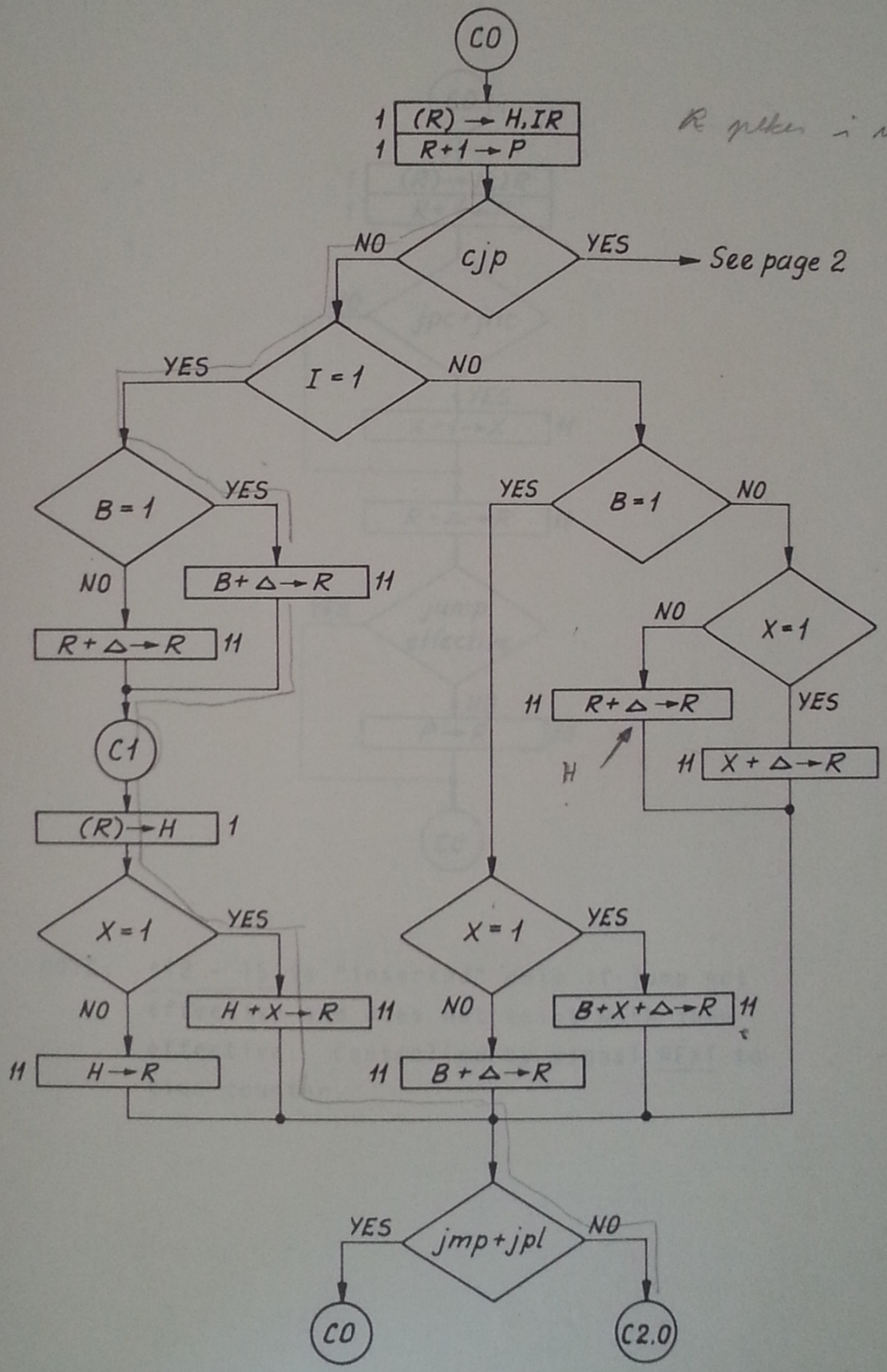
2 FLOW DIAGRAMS

Below follows a list of symbols and abbreviations used in the FLOW DIAGRAMS together with a short explanation.

AW	Anything written	Flip-flops on the Assembler 122
DP	Deposit	
SA	Set Address	
SI	Single Instruction	
ST	Stop	
TR	Tape-Reader	
SR	Source-Register	
DR	Destination-Register	
S	DR - SR (Sum output)	
(R)	The contents of the cell which address is contained in the R-register	
REG (i)	Bit in the register	
Δ	Displacement of the instruction (lower half of H with sign extension)	
A $\leftrightarrow$ G	The A- and the G-register is exchanged	
$T14_0 \rightarrow T14$	Bit 14 in the T-register is complemented	
$2X \rightarrow X$	Left shift of the X-register	
$1/2X \rightarrow X$	Right shift of the X-register	
$T15 \oplus H15 \rightarrow SG1$	Exclusive or of bit 15 of the T- and the H-register. If T15 is the oposite of H15, $1 \rightarrow SG1$	
$T15 \oplus H15_0 \rightarrow SG1$	If T15 is equal to H15, $0 \rightarrow SG1$	
$A0 \cdot 2^{15} + 1/2D \rightarrow D$	Right shift of the D-register with A-register bit 0 shifted into D-register bit 15	

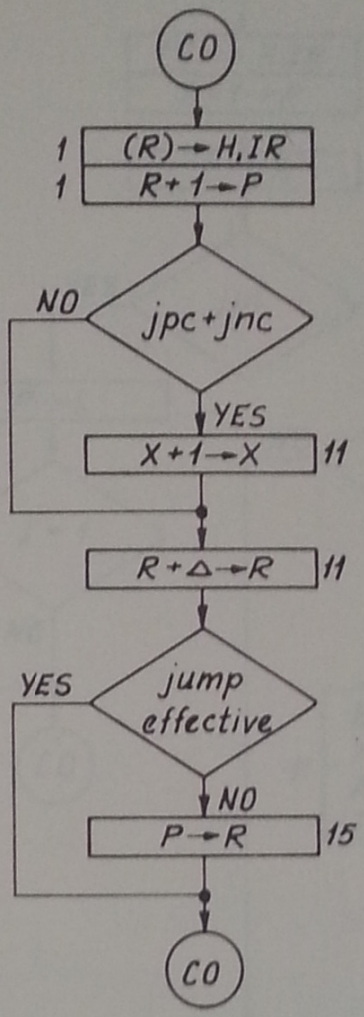
--ooOoo--

A/S NORSK DATA-ELEKTRONIKK	Title: FLOW DIAGRAMS	Drawing no. Page 1



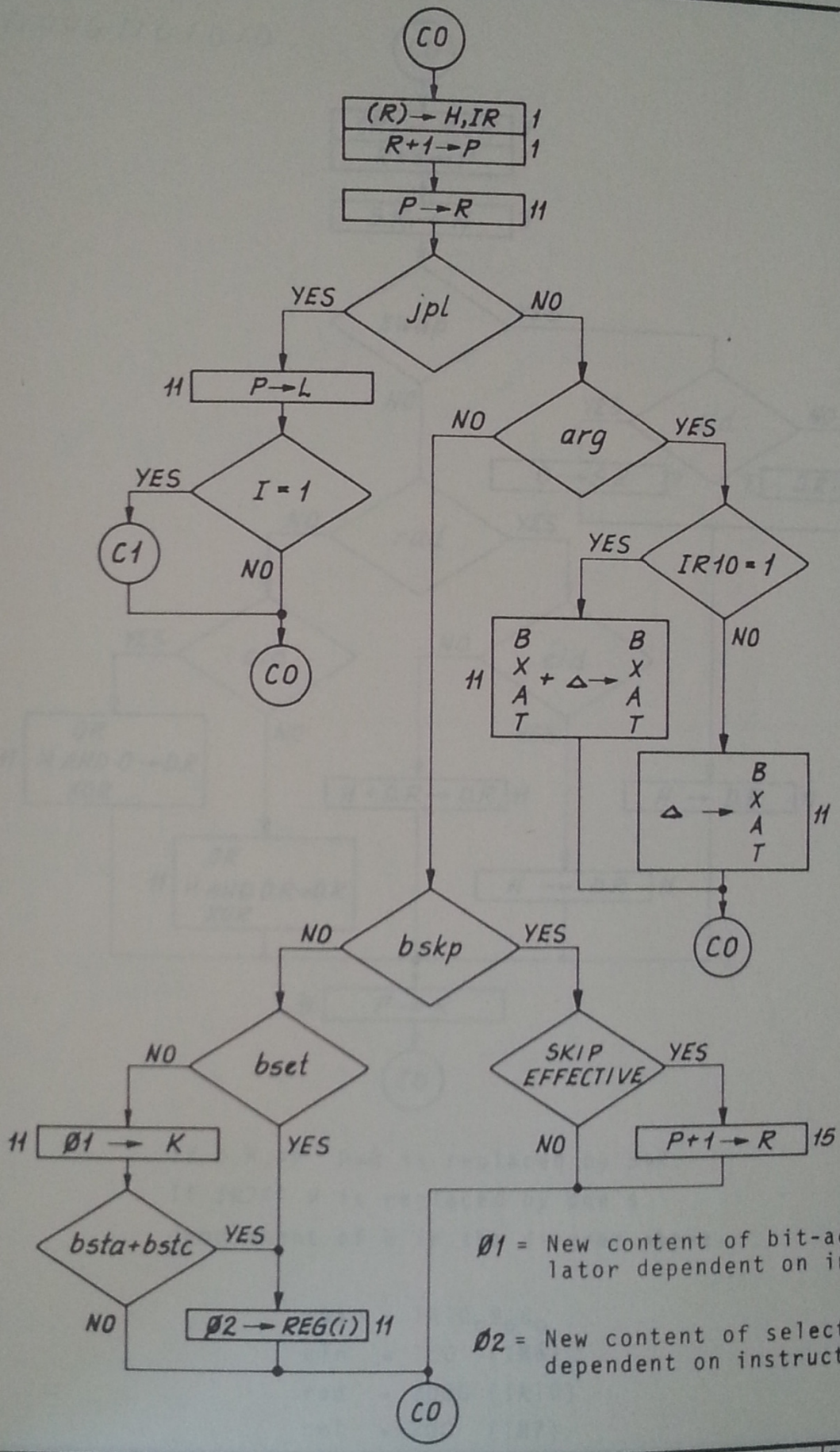
Δ= Lower half of H with sign extension.

DRAWN BY EML	Remarks: ADDRESSING SEQUENCE GO - 5	Replacement for	Date
APPROVED BY		Replaced by	Date
DATE 14.10.69			



NOTE: t12 - 15 is "inserted" only if jump not effective and does not exist with jump effective. Controlled by signal NEXT to time counter.

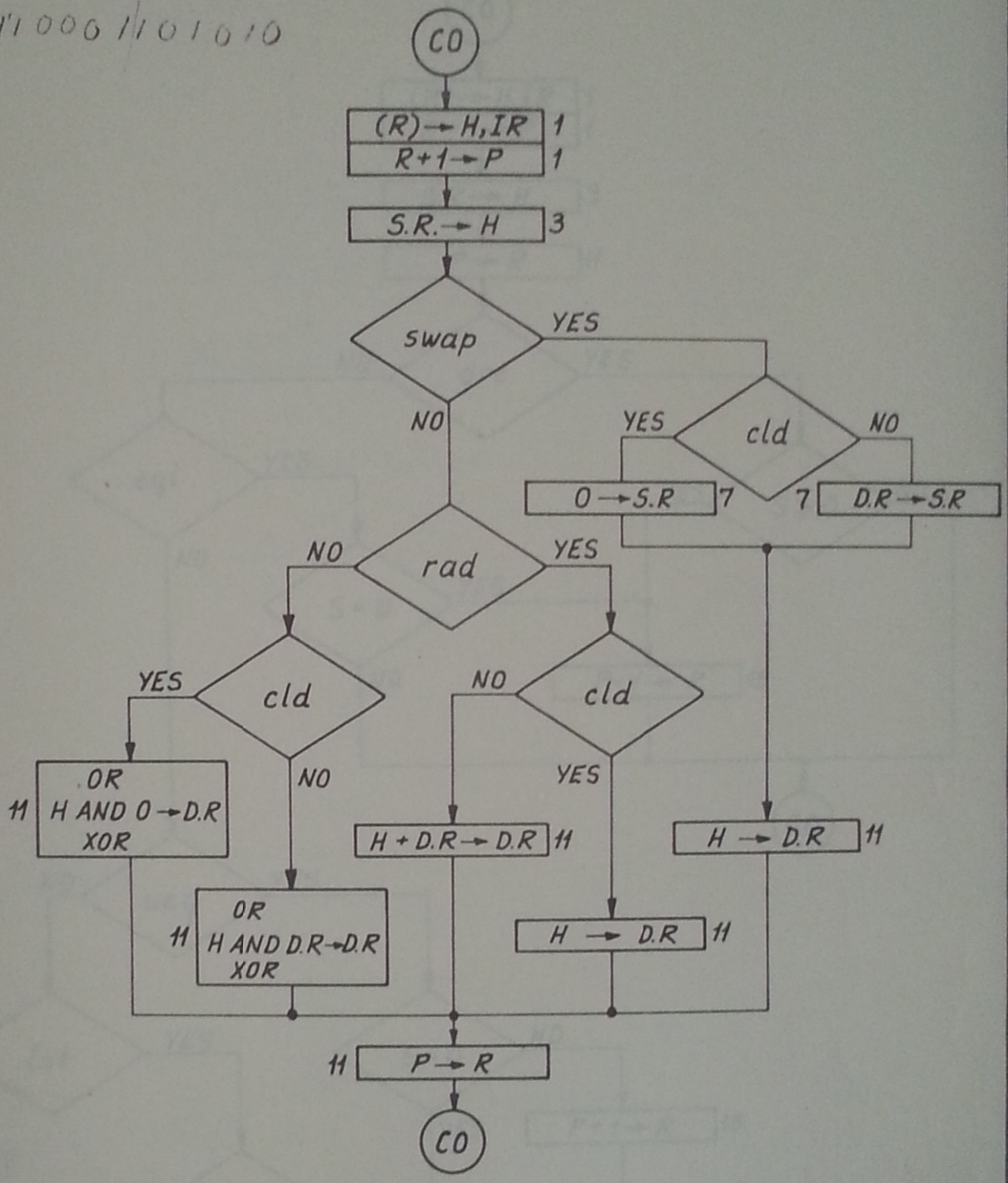
A/S NORSK DATA-ELEKTRONIKK	Title: FLOW DIAGRAMS	Drawing no. Page 3



DRAWN BY EML	Remarks: JPL + ARG + BOP	Replacement for	Date
APPROVED BY		Replaced by	Date
DATE 14.10.69			

A/S NORSK DATA-ELEKTRONIKK	Title: FLOW DIAGRAMS	Drawing no. Page 4

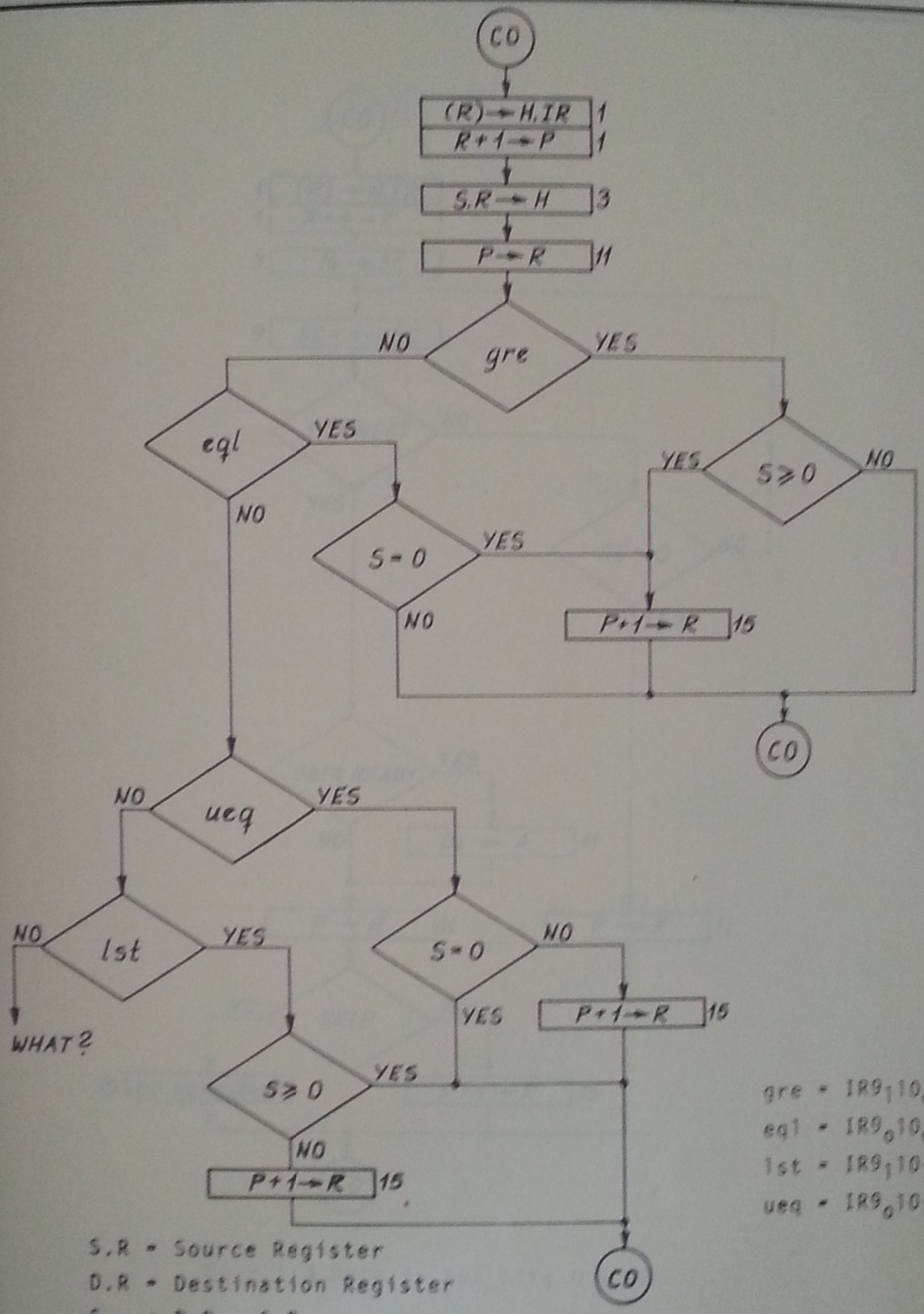
1100111006 1101010



If D.R.=P P→R is replaced by S→R.
If IR7=1 H is replaced by one's complement of H in the diagram above.

swap = $IR10_0 9_0 8_0$
cld = 100 (IR6)
rad = 2000 (IR10)
cml = 200 (IR7)

DRAWN BY EML	Remarks	Replacement for	Date
APPROVED BY	R O P	Replaced by	Date
DATE 15.10.69			



gre = $IR9_1 10_0$

eql = $IR9_0 10_0$

lst = $IR9_1 10_1$

ueq = $IR9_0 10_1$

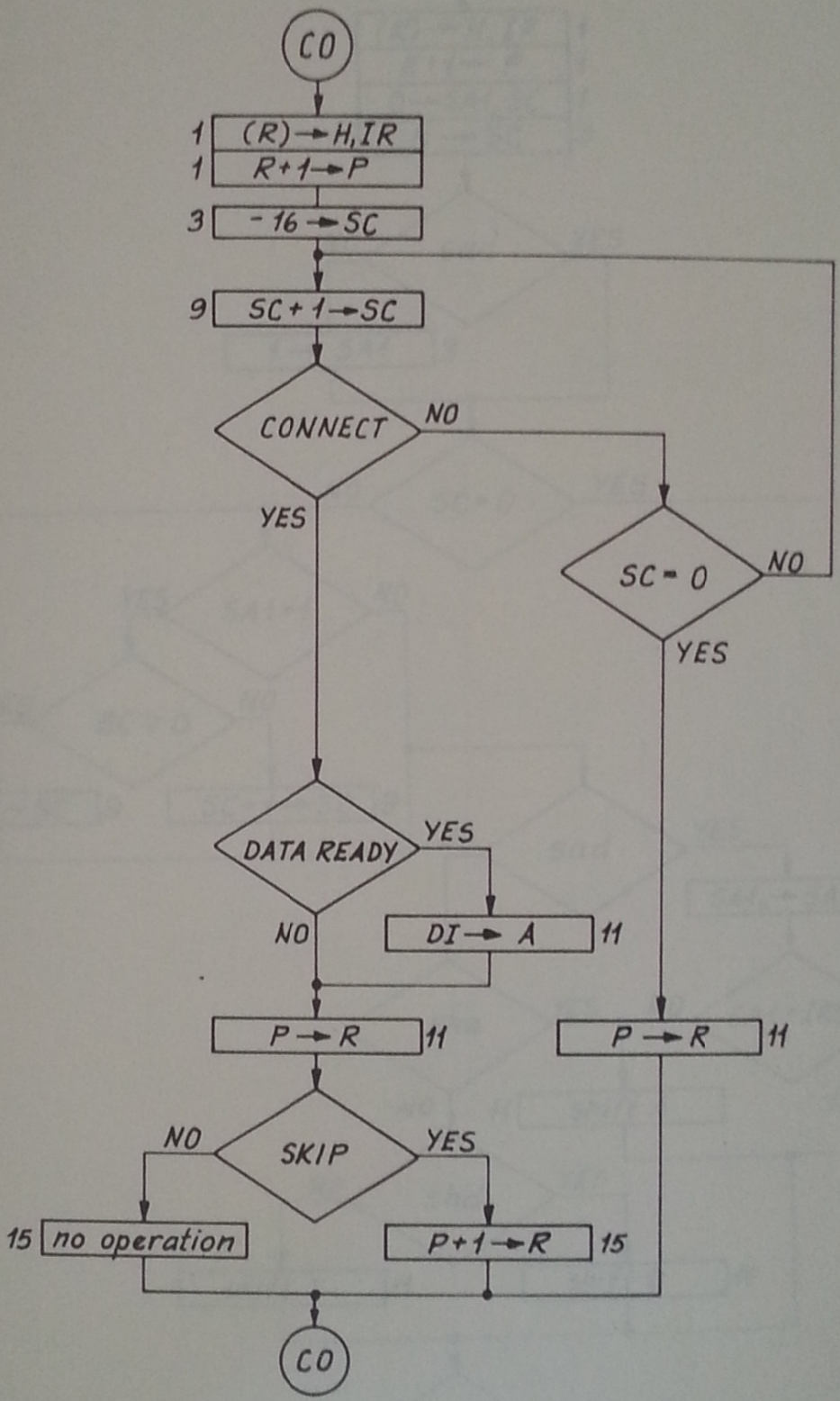
S.R = Source Register

D.R = Destination Register

S = D.R - S.R

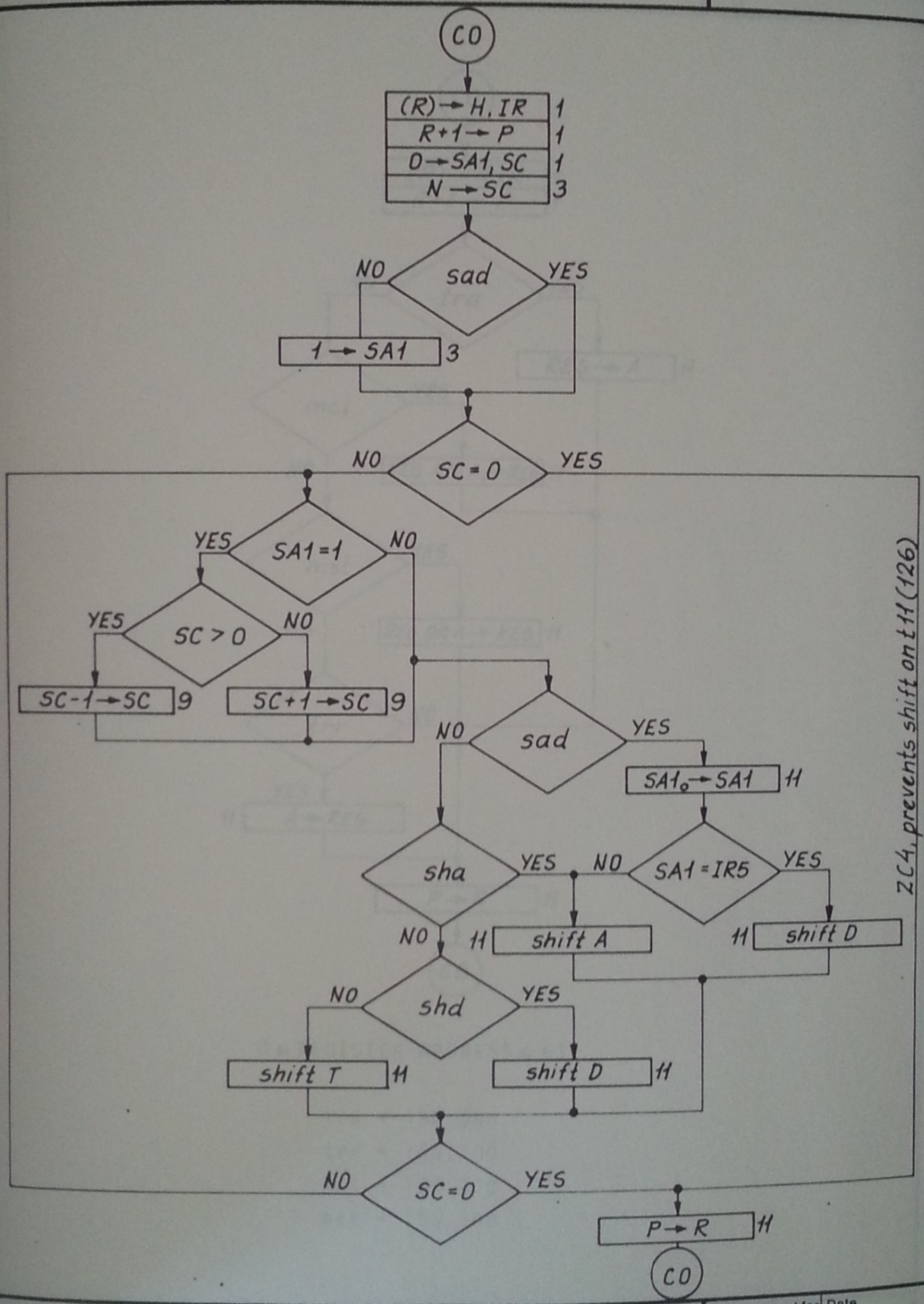
See also note page 2

DRAWN BY EML	Remarks	Replacement for / Date
APPROVED BY	S K P	Replaced by / Date
DATE 16.10.69		

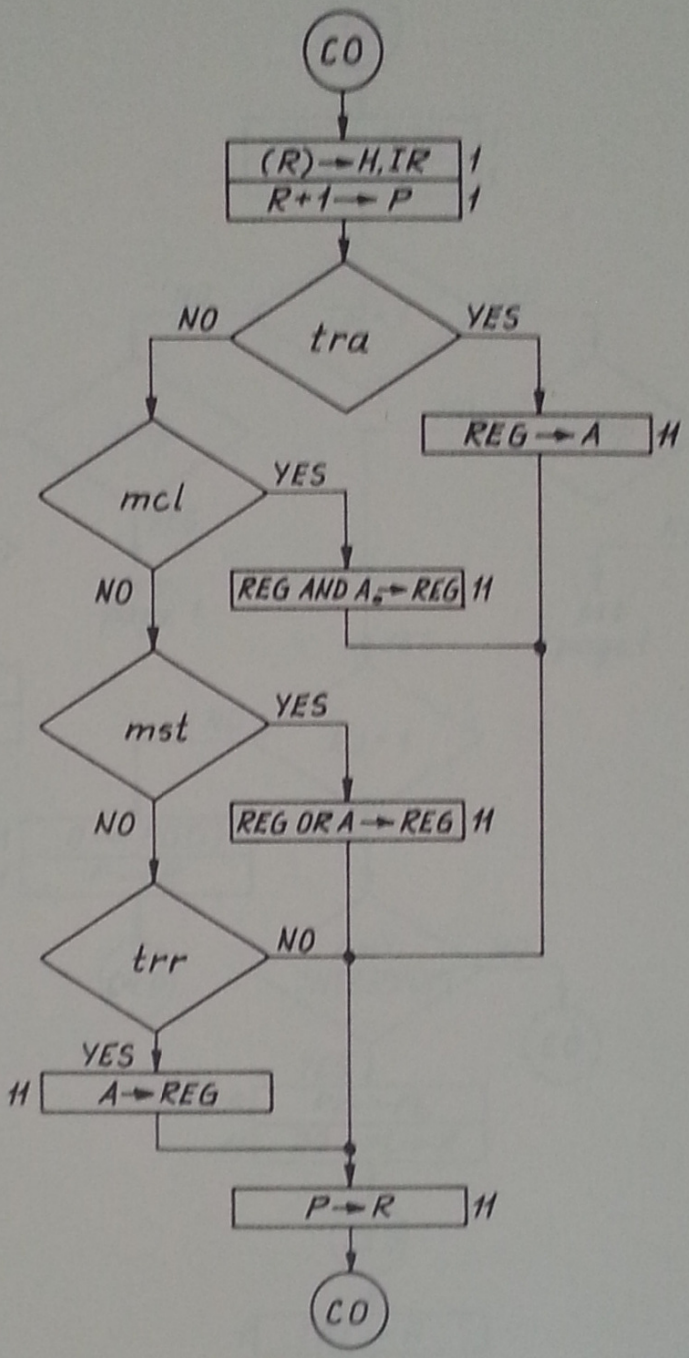


DI = I/O Data in bus bits 0 - 15.

A/S NORSK DATA-ELEKTRONIKK	Title: FLOW DIAGRAMS	Drawing no. Page 7

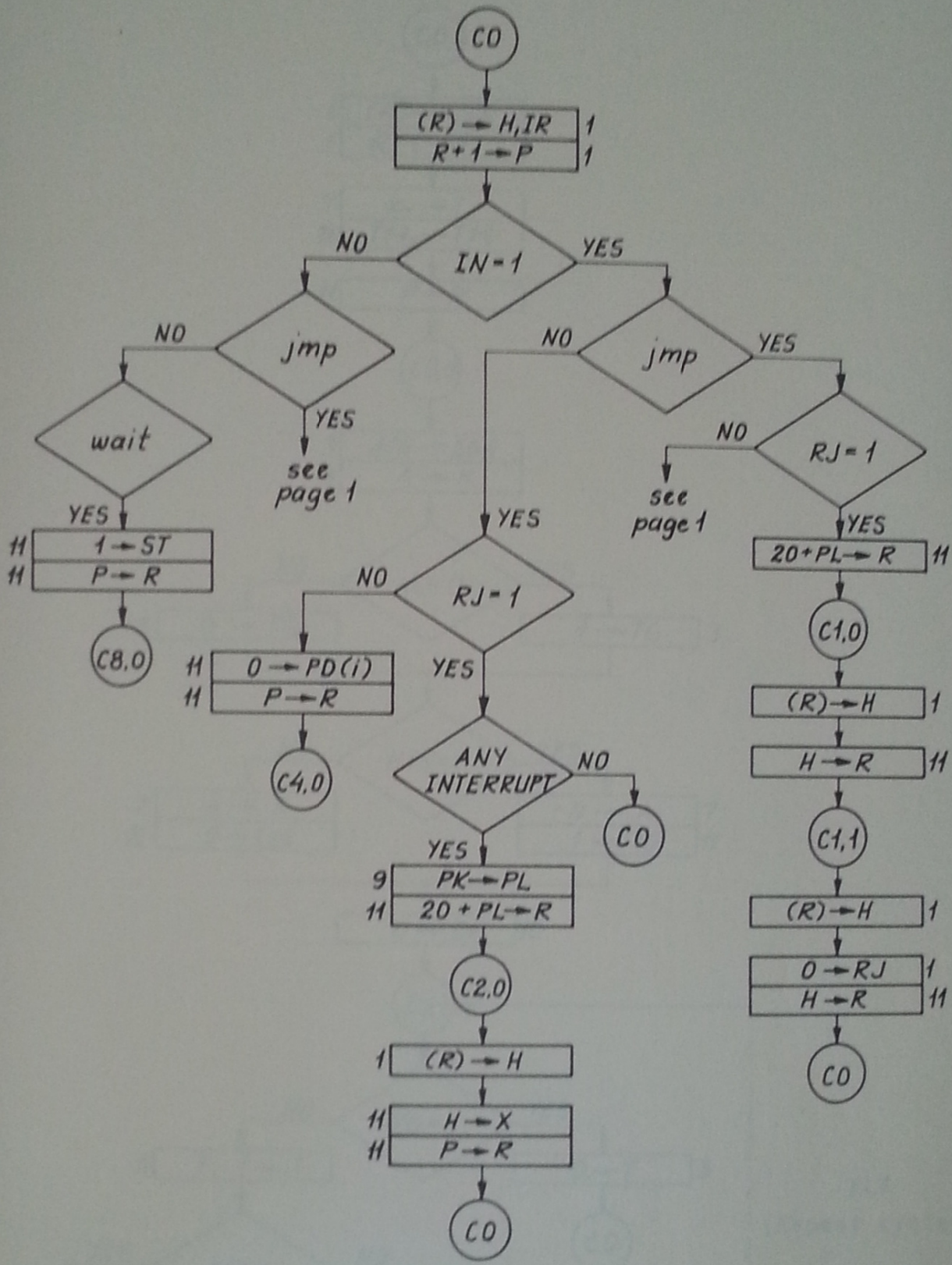


DRAWN BY EML	Remarks: S H T	Replacement for	Date
APPROVED BY		Replaced by	Date
DATE 5.11.69			



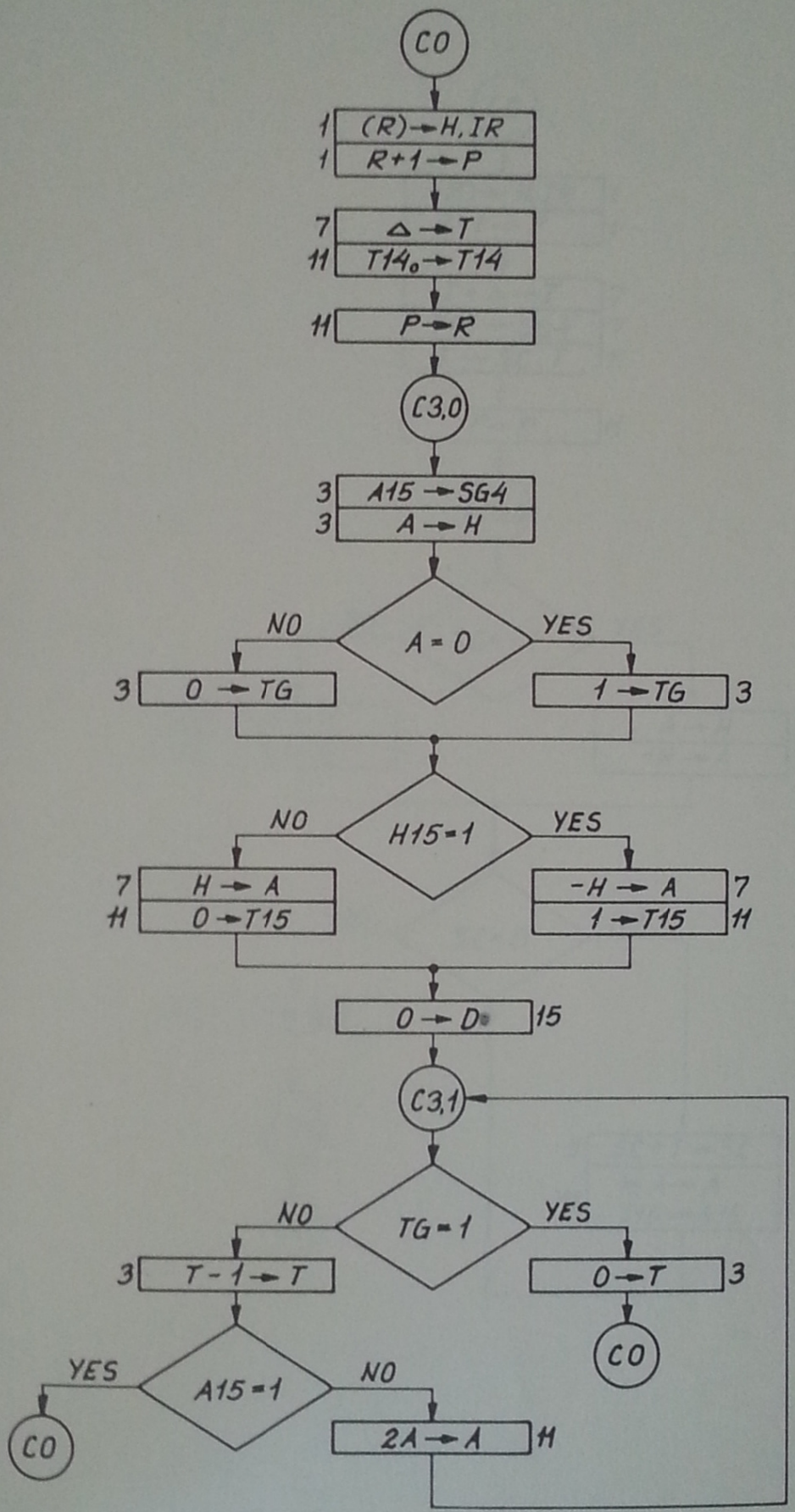
0 ≤ REGISTER ADDRESS ≤ 63

tra = 150,000
trr = 150,100
mcl = 150,200
mst = 150,300

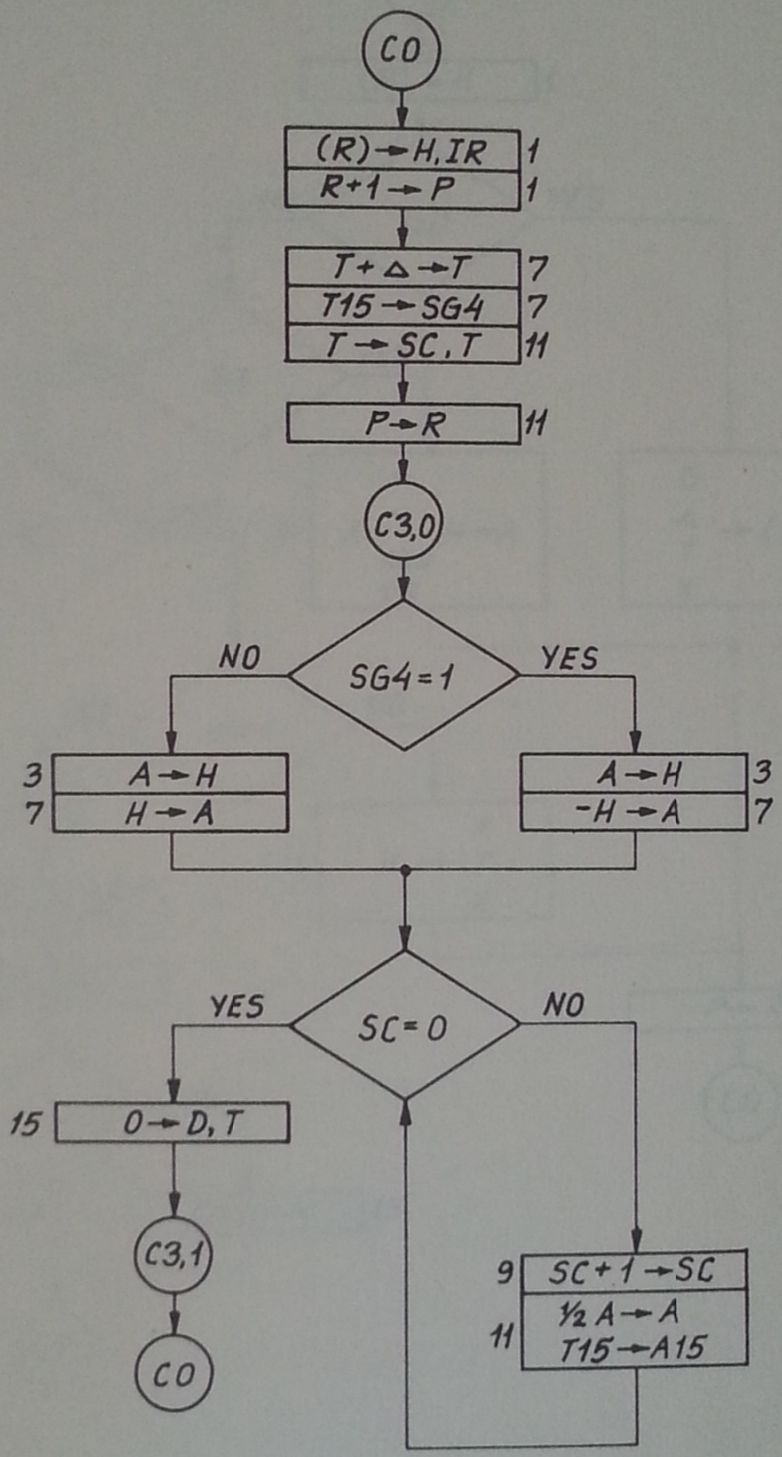


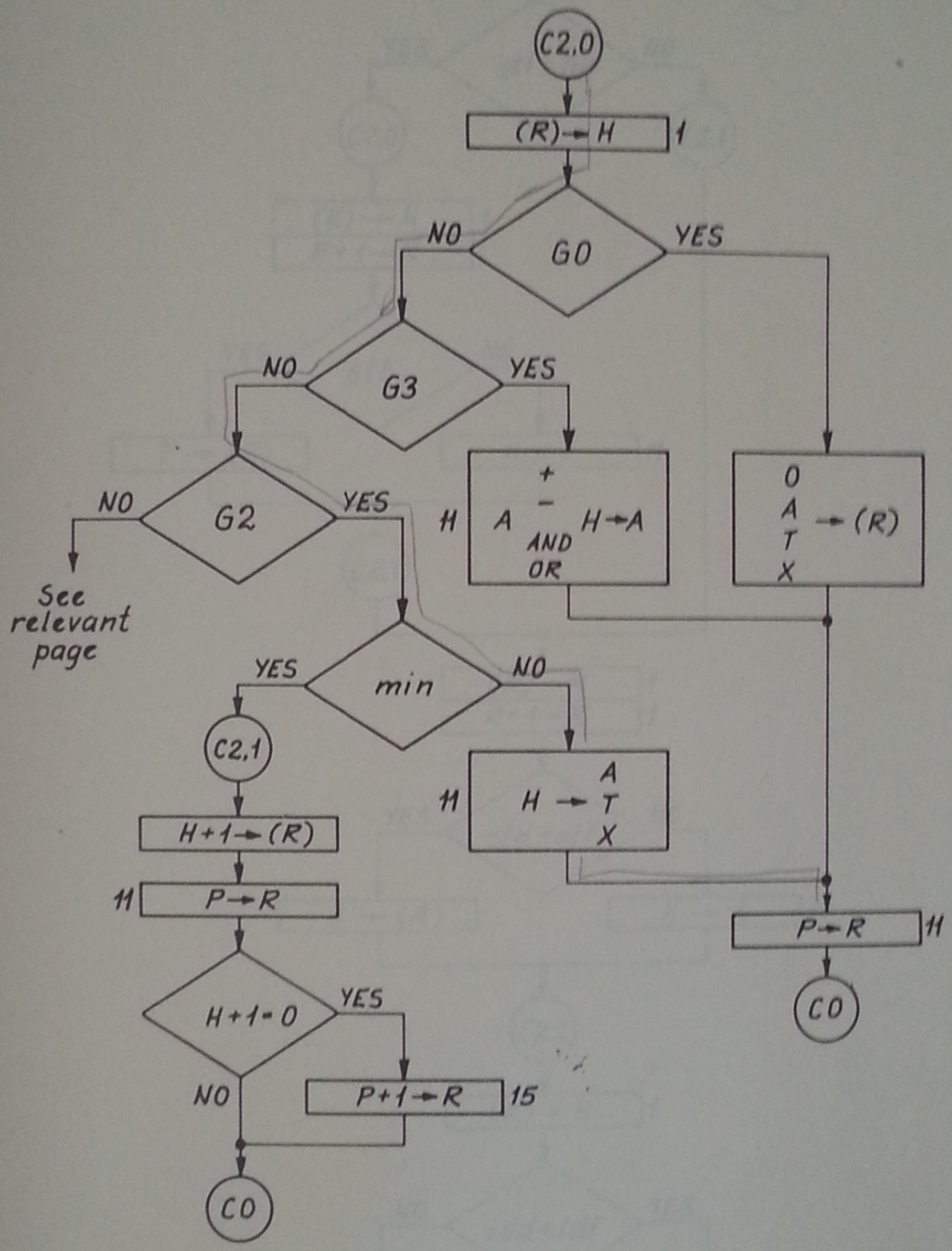
△ = displacement

DRAWN BY EML	Remarks	Replacement for	Date
APPROVED BY	WAIT + JMP • IN		
DATE 15.10.69		Replaced by	Date



FL4
(Repeat cycle)


CO
(R)→H,IR 1
R+1→P 1
T+Δ→T 7
T15→SG4 7
T→SC,T 11
P→R 11
C3,0
NO
SG4=1
YES
3 A→H
7 H→A
A→H 3
-H→A 7
YES
SC=0
NO
15 0→D,T
C3,1
CO
9 SC+1→SC
11 ½A→A
T15→A15

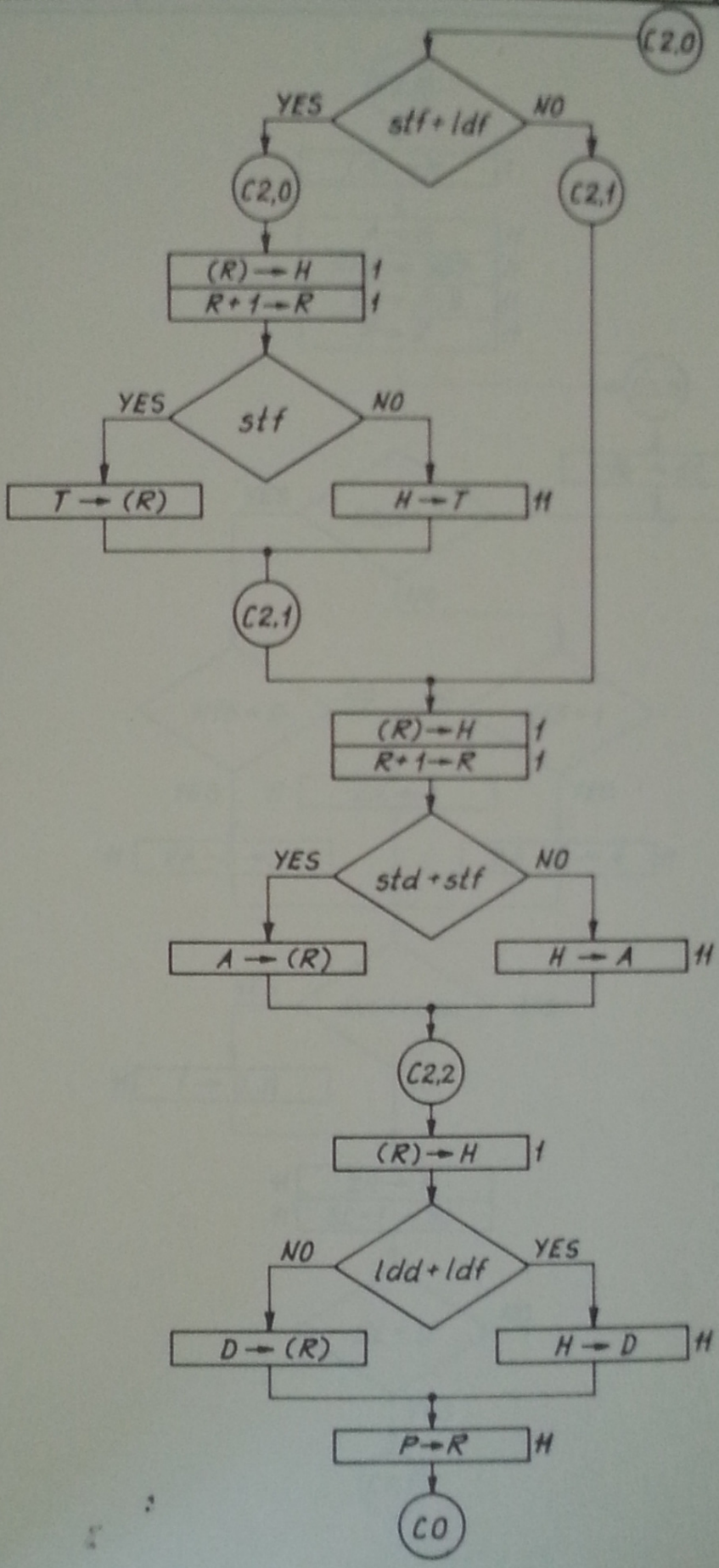


G0 = stz + sta + stt + stx

G2 = min + lda + ldt + ldx

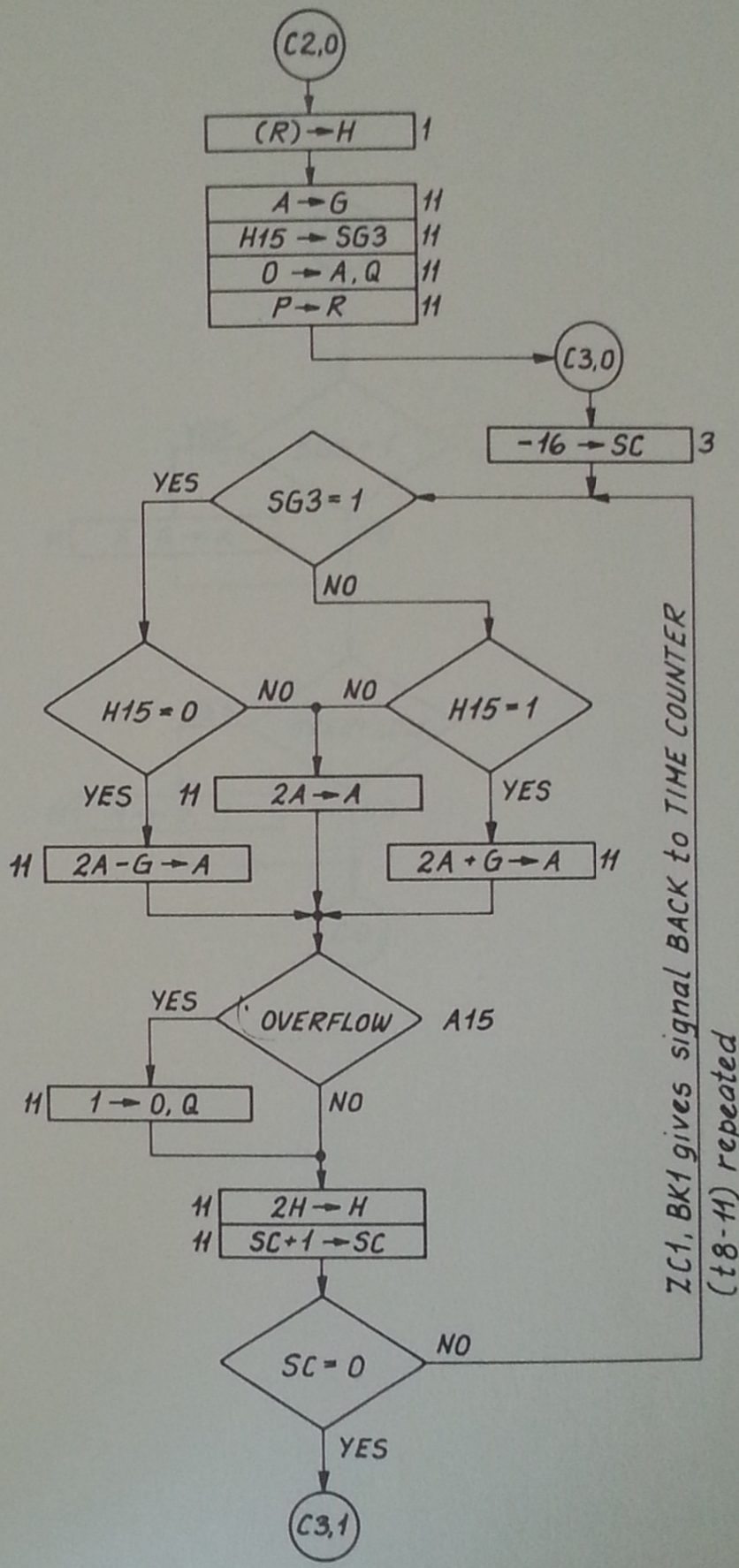
G3 = add + sub + and + ora

DRAWN BY EML	Remarks	Replacement for / Date
APPROVED BY	G0 + 2 + 3	Replaced by / Date
DATE 16.10.69		

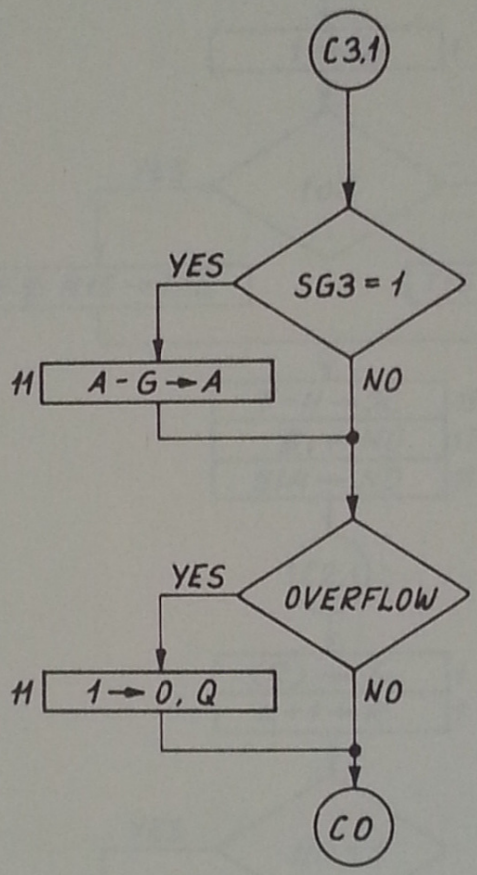


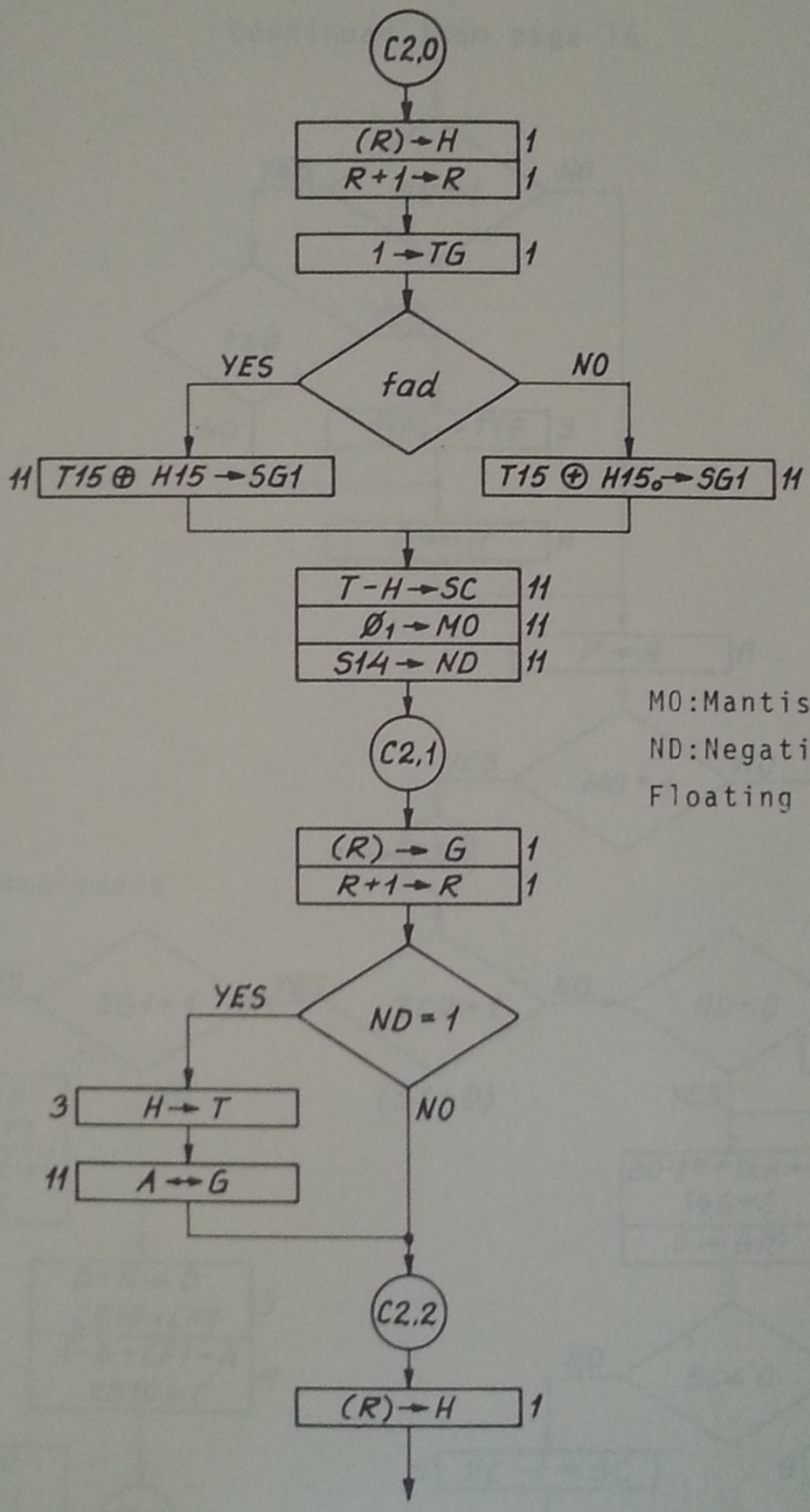
DRAWN BY EML	Remarks	Replacement for	Date
APPROVED BY	STD + STF + LDD + LDF	Replaced by	Date
DATE 16.10.69			

7450



DRAWN BY EML	Remarks	Replacement for	Date
APPROVED BY			
DATE 16.10.69	M P Y	Replaced by	Date


C3.1
SG3 = 1
YES
NO
A - G → A
OVERFLOW
YES
NO
1 → 0, Q
C0



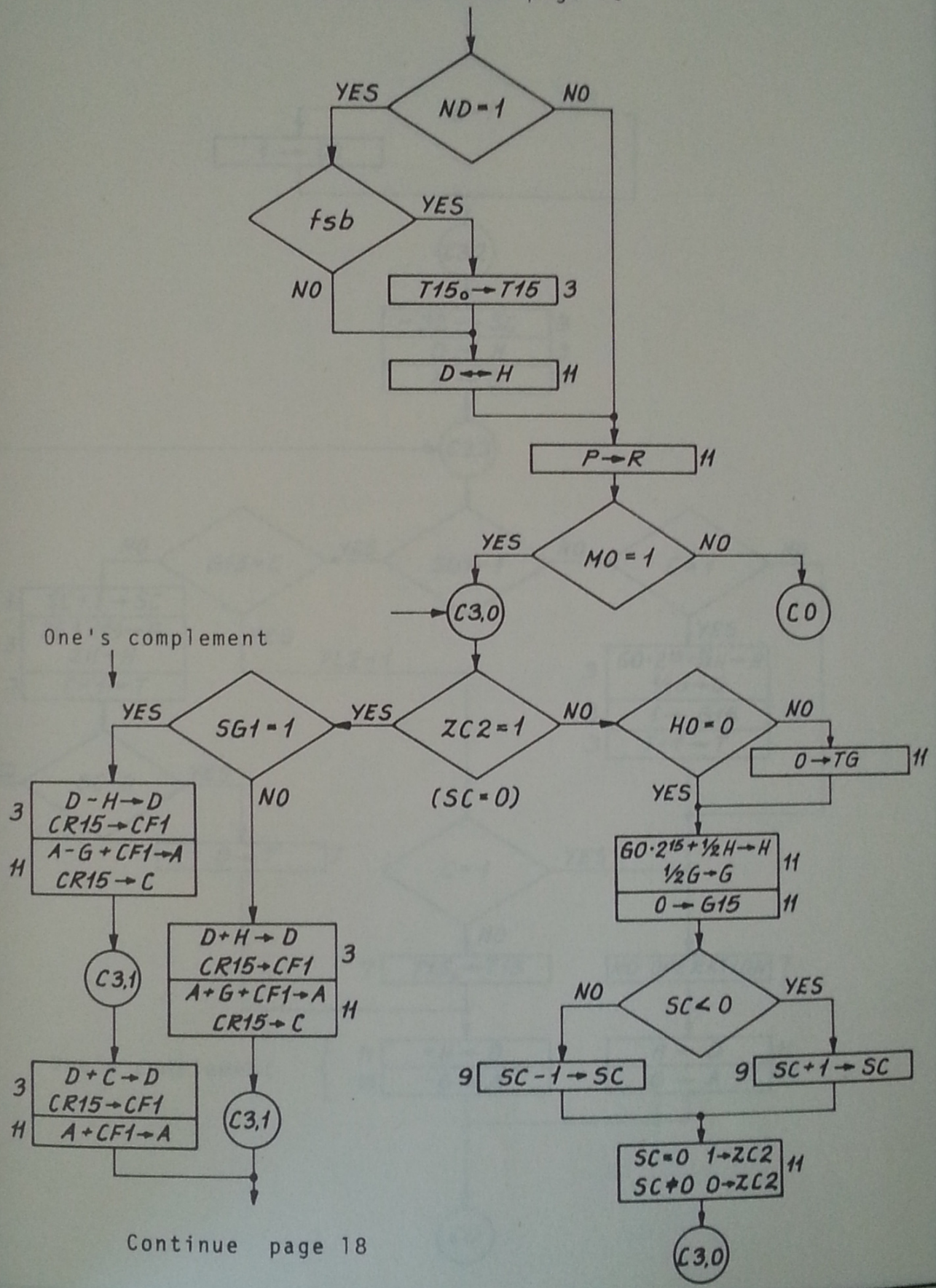
MO:Mantissa overlap
ND:Negative differance
Floating control 134

Continue page 17

$$Ø_1 = S14_1 13_1 12_1 11_1 10_1 9_1 8_1 7_1 6_1 5_1 + S14_0 13_0 12_0 11_0 10_0 9_0 8_0 7_0 6_0 5_0$$

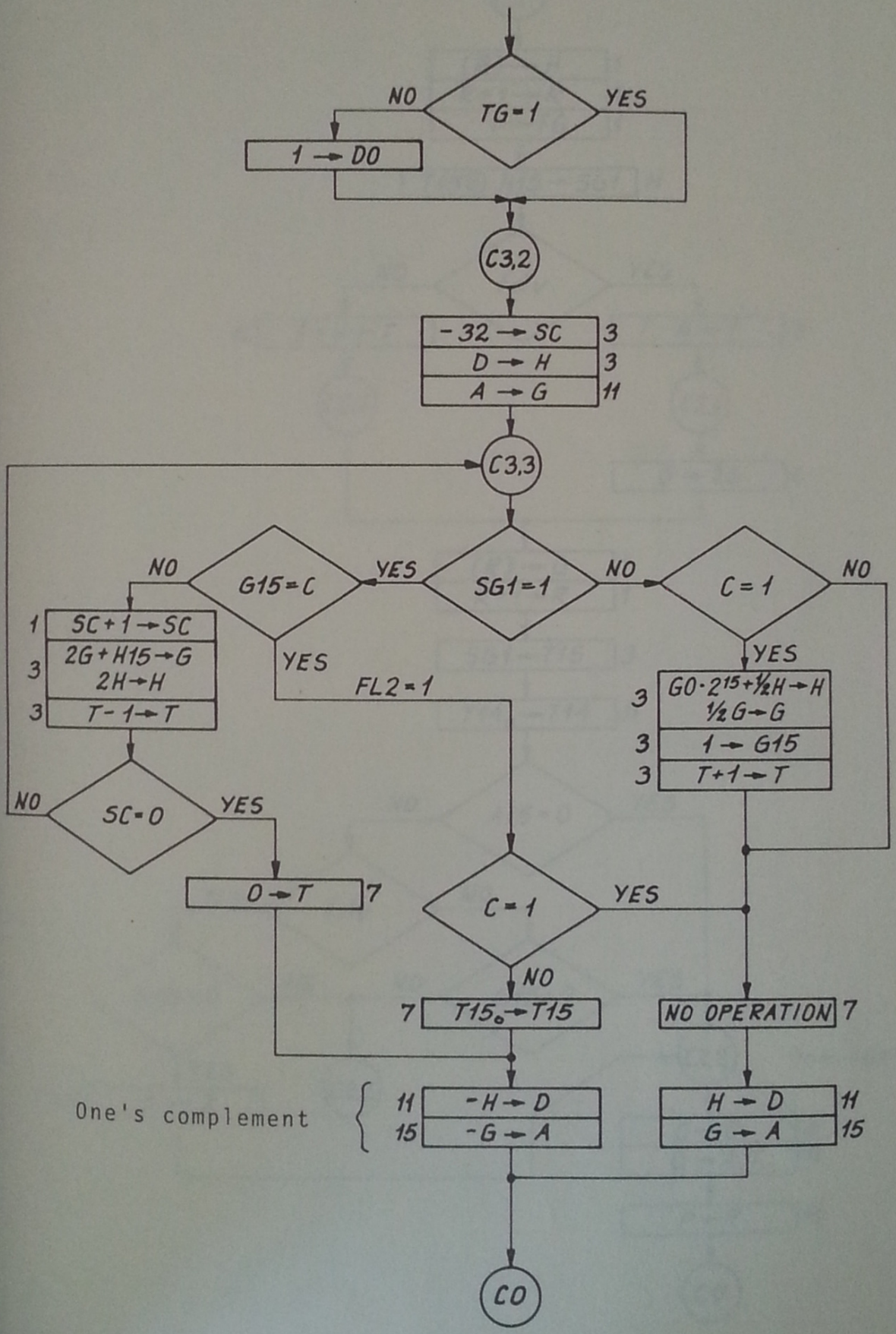
): (T - H) < 32

Continued from page 16

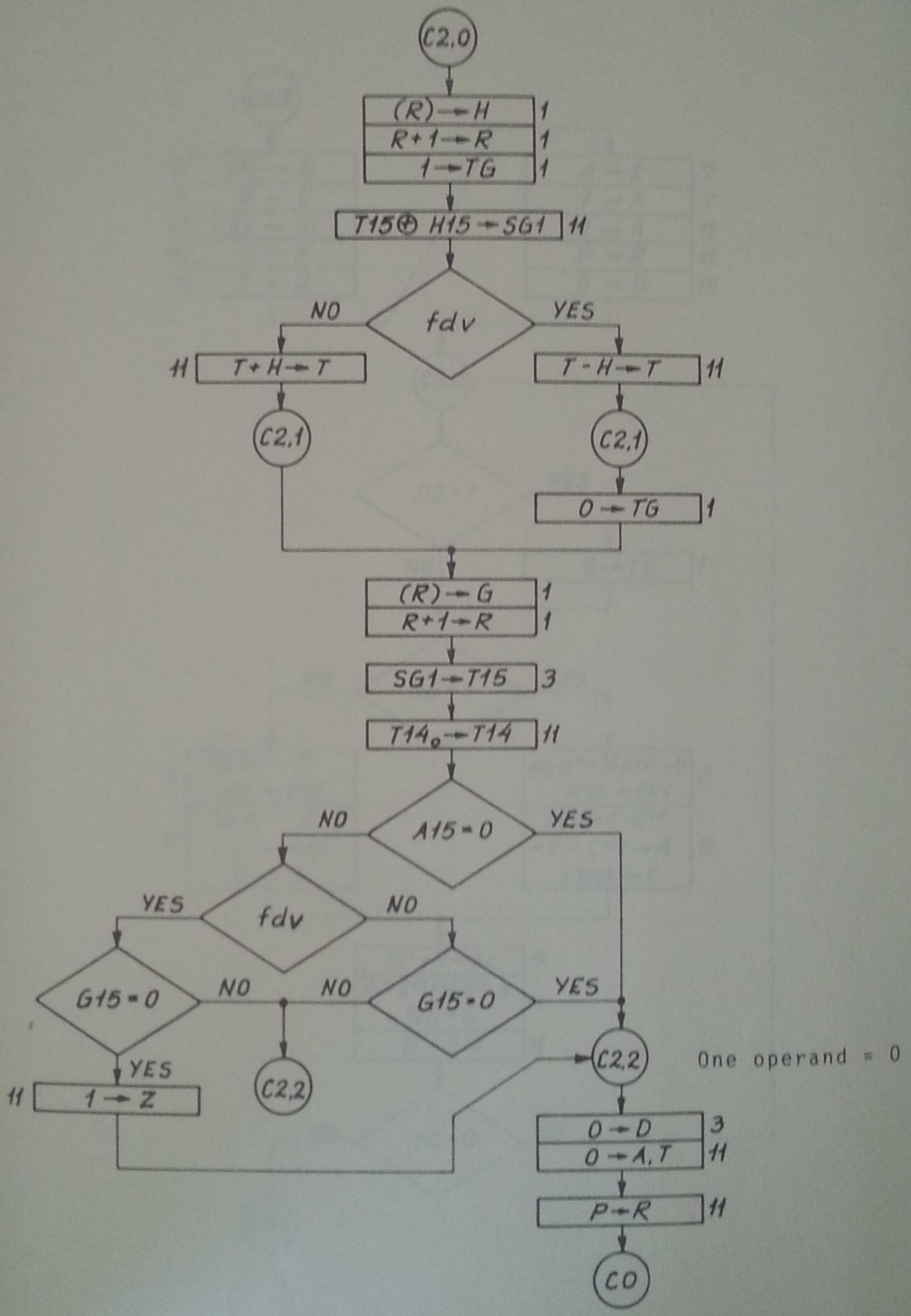


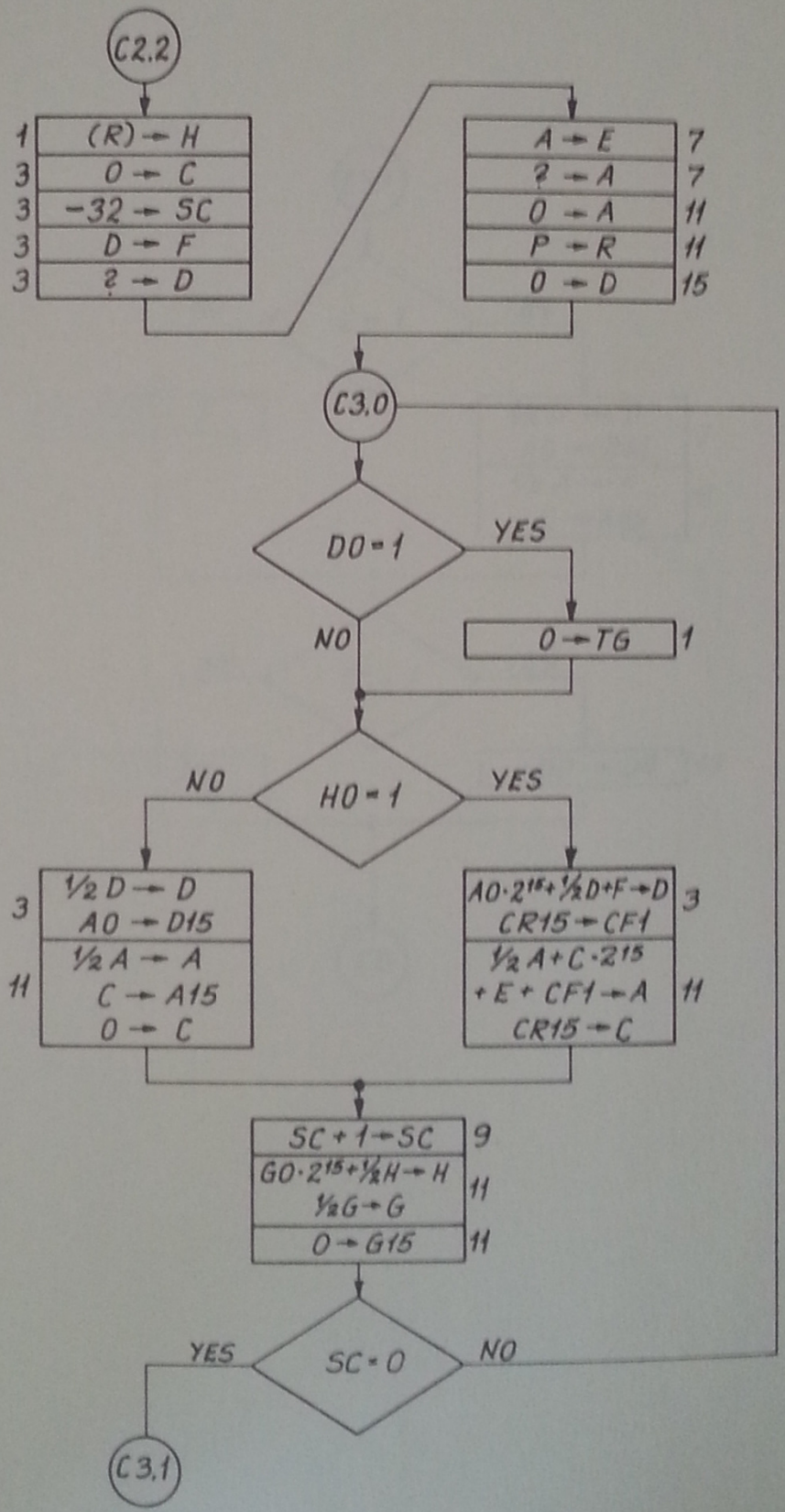
Continue page 18

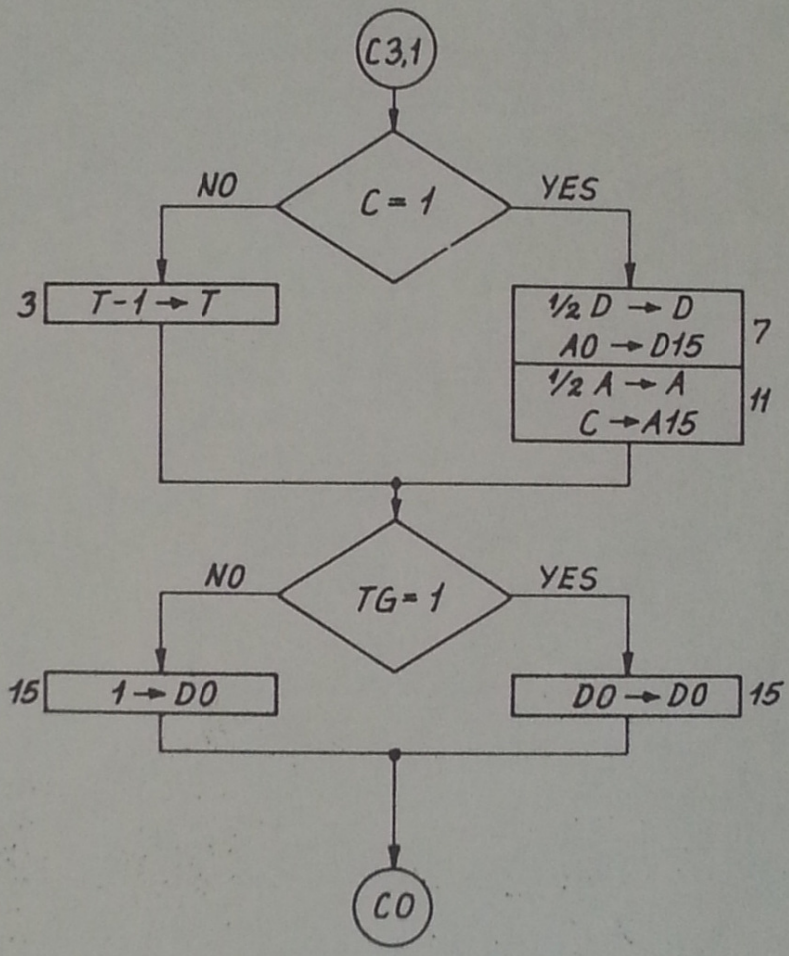
Continued from page 17



DRAWN BY EML	Remarks	Replacement for	Date
APPROVED BY	FAD + FSB		
DATE 17.10.69		Replaced by	Date


C2,0
(R) → H 1
R+1 → R 1
1 → TG 1
T15 ⊕ H15 → SG1 11
fdv
NO
YES
11 T+H → T
T-H → T 11
C2,1
C2,1
0 → TG 1
(R) → G 1
R+1 → R 1
SG1 → T15 3
T14₀ → T14 11
A15 = 0
NO
YES
fdv
YES
NO
G15 = 0
NO
NO
G15 = 0
YES
YES
11 1 → Z
C2,2
C2,2
0 → D 3
0 → A,T 11
P → R 11
CO
One operand = 0


C2.2
1 (R) → H
3 0 → C
3 -32 → SC
3 D → F
3 ? → D
A → E 7
? → A 7
0 → A 11
P → R 11
0 → D 15
C3.0
D0 = 1
YES
NO
0 → TG 1
H0 = 1
NO
YES
3 ½ D → D
A0 → D15
11 ½ A → A
C → A15
0 → C
A0·2^15 + ½D + F → D 3
CR15 → CF1
½ A + C·2^15 + E + CF1 → A 11
CR15 → C
SC + 1 → SC 9
G0·2^15 + ½H → H 11
½G → G
0 → G15 11
SC = 0
YES
NO
C3.1


C3,1
C = 1
NO
YES
3
T-1 → T
½ D → D
A0 → D15
7
½ A → A
C → A15
11
TG = 1
NO
YES
15
1 → D0
D0 → D0
15
C0