PARALLELL ADDISJON.

Parallell-addisjon er mye raskere enn serie-addisjon og er derfor brukt i større computere (high-speed).
Ved serie-addisjon blit bit-parene lagt inn i adderen på serieform, dvs. det vil ta en bestemt tid å mate adderen med data. I tillegg kommer forsinkelsen i selve adderen.
Ved parallell-addisjon blir addendene presentert adder-inngangene samtidig uten tap av tid.

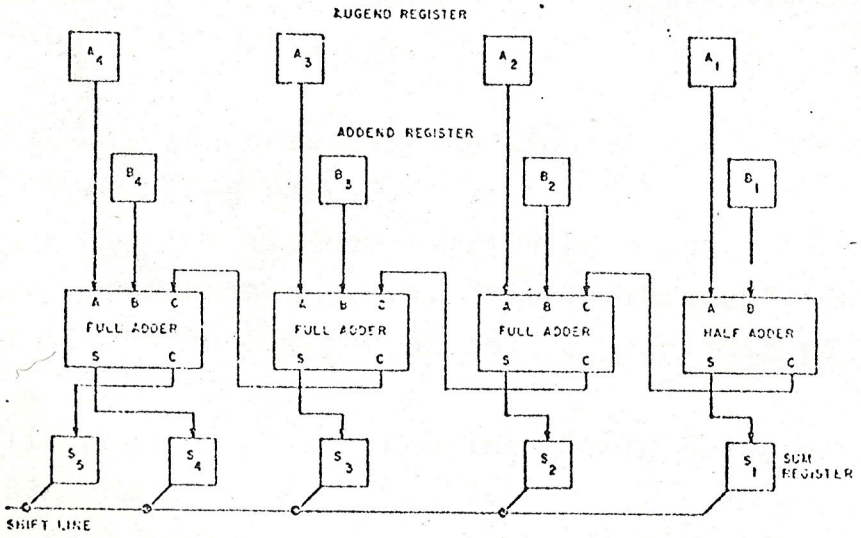


Fig. 1.

Fig. 1 viser en parallell-adder blokkskjematisk. Det trengs en adder for hvert bit-par.
Hvorfor er adderen for det minst signifikante bit-paret av typen half-adder?

Svar: _____

_____

Etter at summen s er formet i half-adderen, blir mentet transportert til menteinngangen i neste adder og addert sammen med neste bit-par osv. Summen blir til slutt kjørt parallelt inn på et sumregister.
Vi bør merke oss at sluttsummen ikke foreligger før mentet fra underliggende operasjon opptrer. Mentene må transporteres,

ett av gangen, fra adder til adder inntil mentet opphører. Denne menteoverføringen kalles "RIPPLE-THROUGH". La oss si at de binære tallene 0001 og 1111 skal summeres.
Den riktige summen vil ikke foreligge før fire menter har "rippled through" logikken.
I tiden fra addendene mates inn og til den riktige sluttsum foreligger, vil det opptre flere summer før mentet når fram til siste adder. Tiden som går med er altså summen av den tid som trengs til å forme en enkelt sum, og den tid som trengs for en "ripple through".

En annen type addere er "The simultaneous-carry adder" også kalt "look ahead carry".
Her blir mentet til hvert trinn presentert samtidig slik at vi unngår forsinkelsen p.g.a. "ripple through". Dette vil igjen si at summen foreligger korrekt og samtidig på alle utgangene uten mellomstadier.
En slik adder har vi i den integrerte brikken SN7483A som er vist i fig. 2.
Denne full-adderen adderer to 4-bit ord, men to addere kan seriekoples slik at addisjon av 8-bit ord kan realiseres.
Vi får da en "ripple through" fra den ene adderen til den andre.

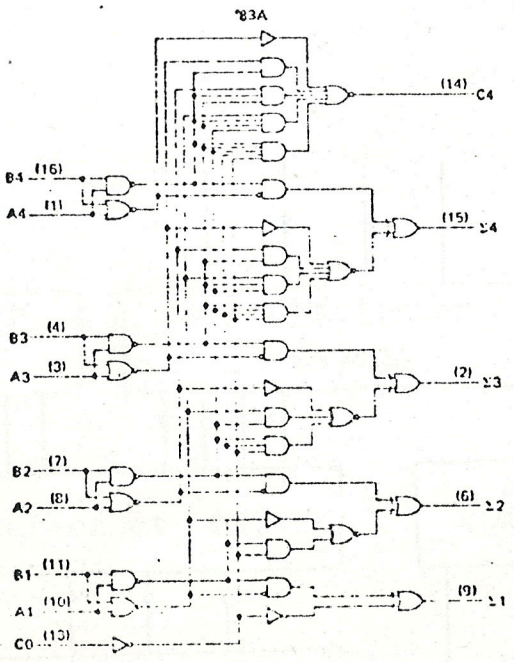
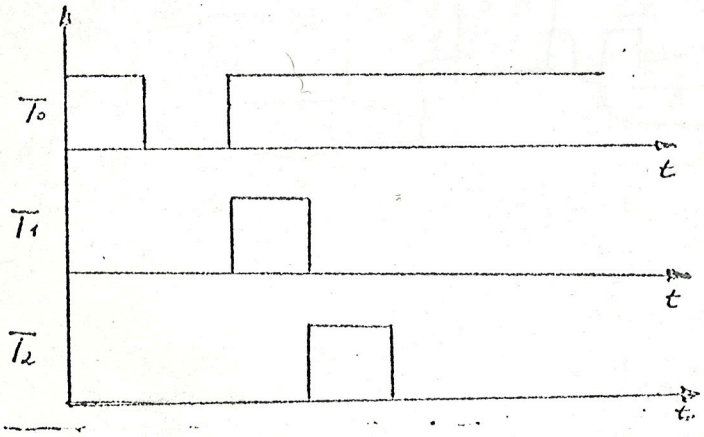


Fig. 2.

Kopl opp adderen etter blokkskjemaet i fig. 3. Bruk SN7496 som "augend-, addend- og sumregister".
Augend- og addendregisteret skal mates parallelt. Konstruer også en enhet som "timer" operasjonene i adderen. Dvs. en enhet som sørger for at augend- og addendregistrene først fylles med data og deretter gir signal til sumregisteret om å lagre summen.
Konstruksjonen tegnes på neste side.

Timing-enheten skal gi tre pulser ut, se timing-diagram.
Den første pulsen, T_0, skal nullstille de tre registrene, T_1 skal mate addend- og augendregistrene med data, mens T_2 skal kjøre resultatet inn i sumregisteret.



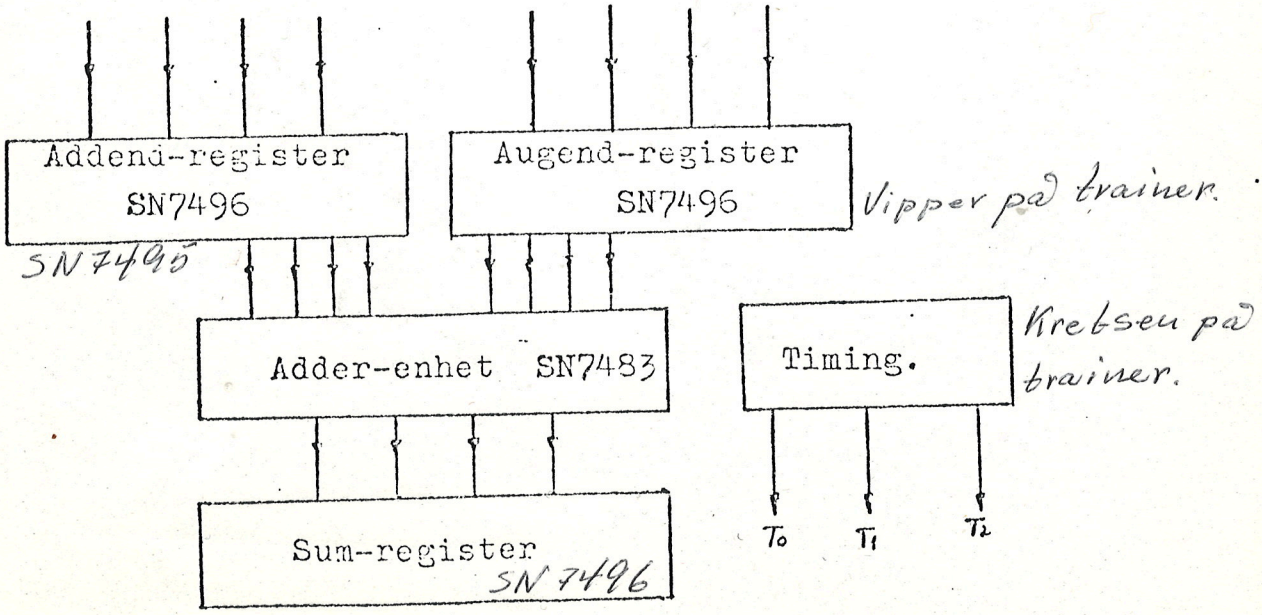
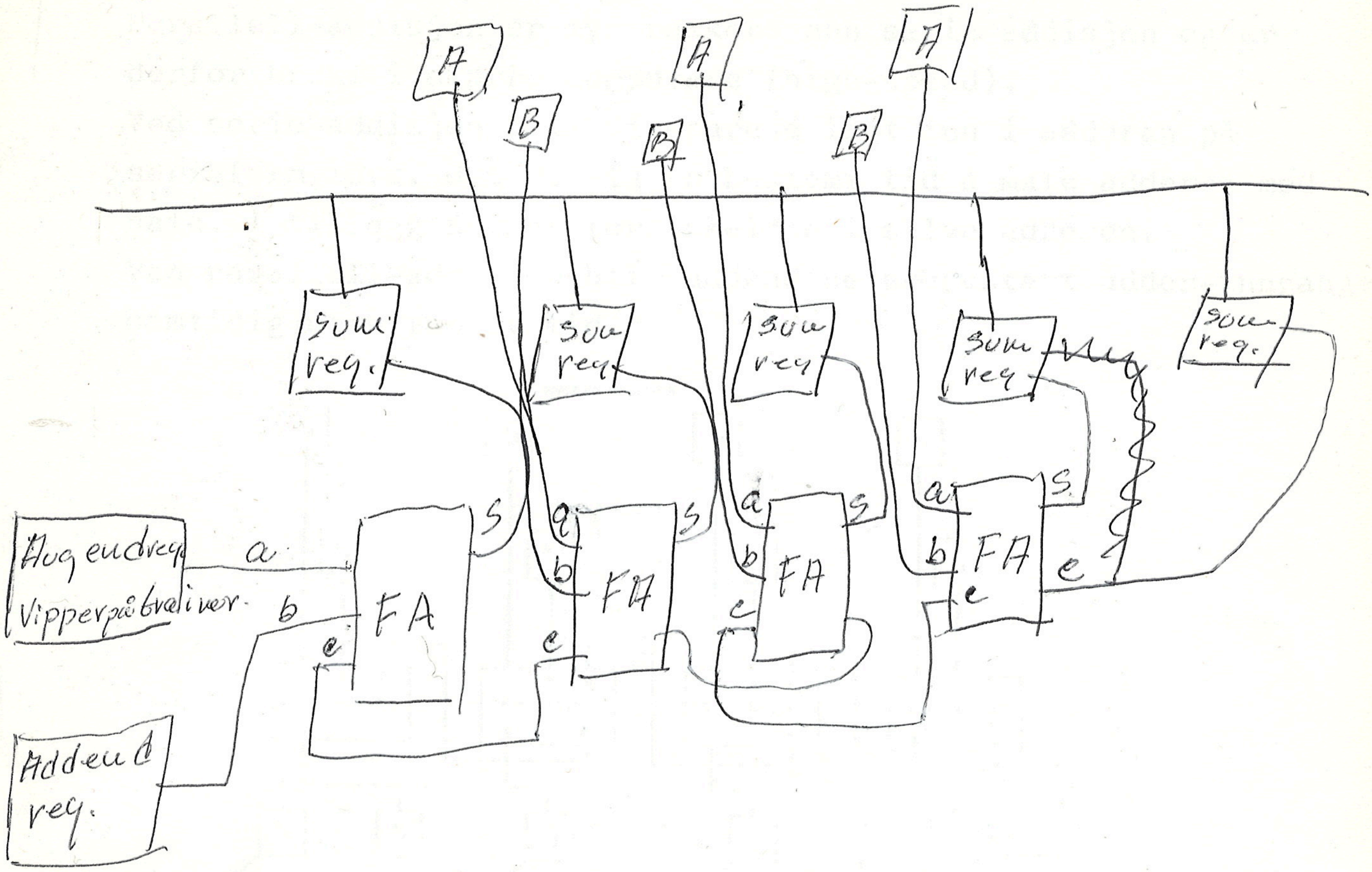
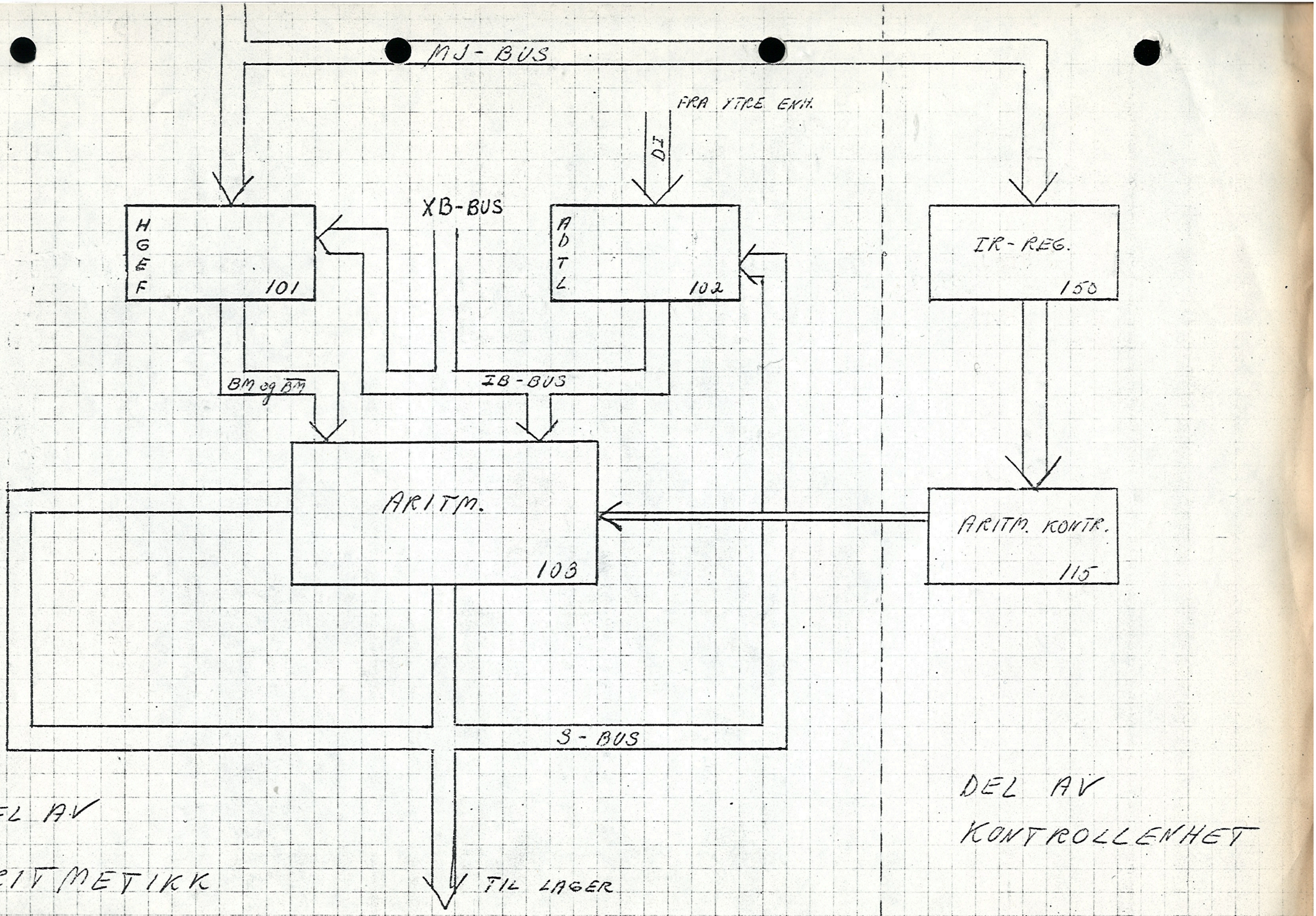
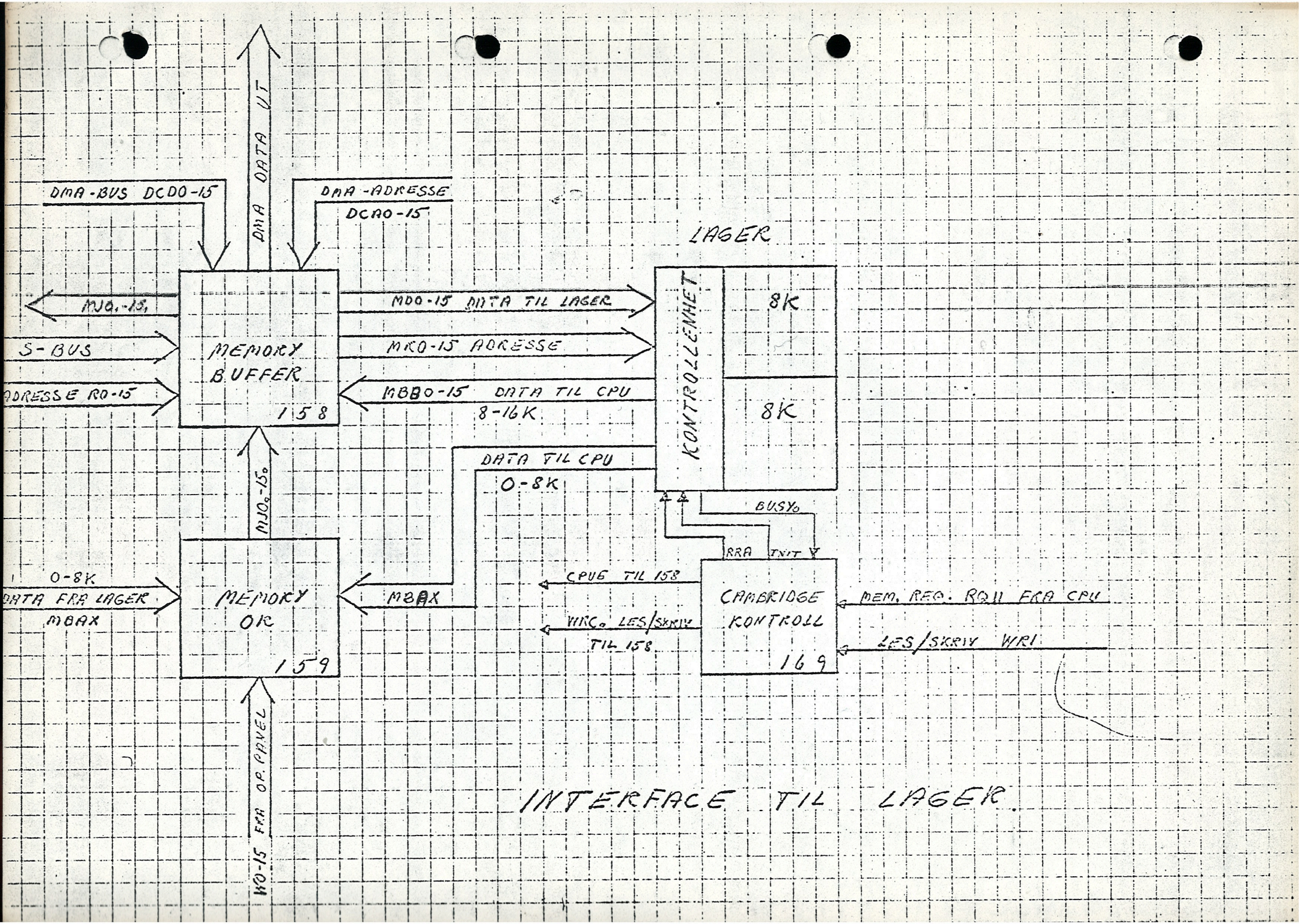


Fig. 3

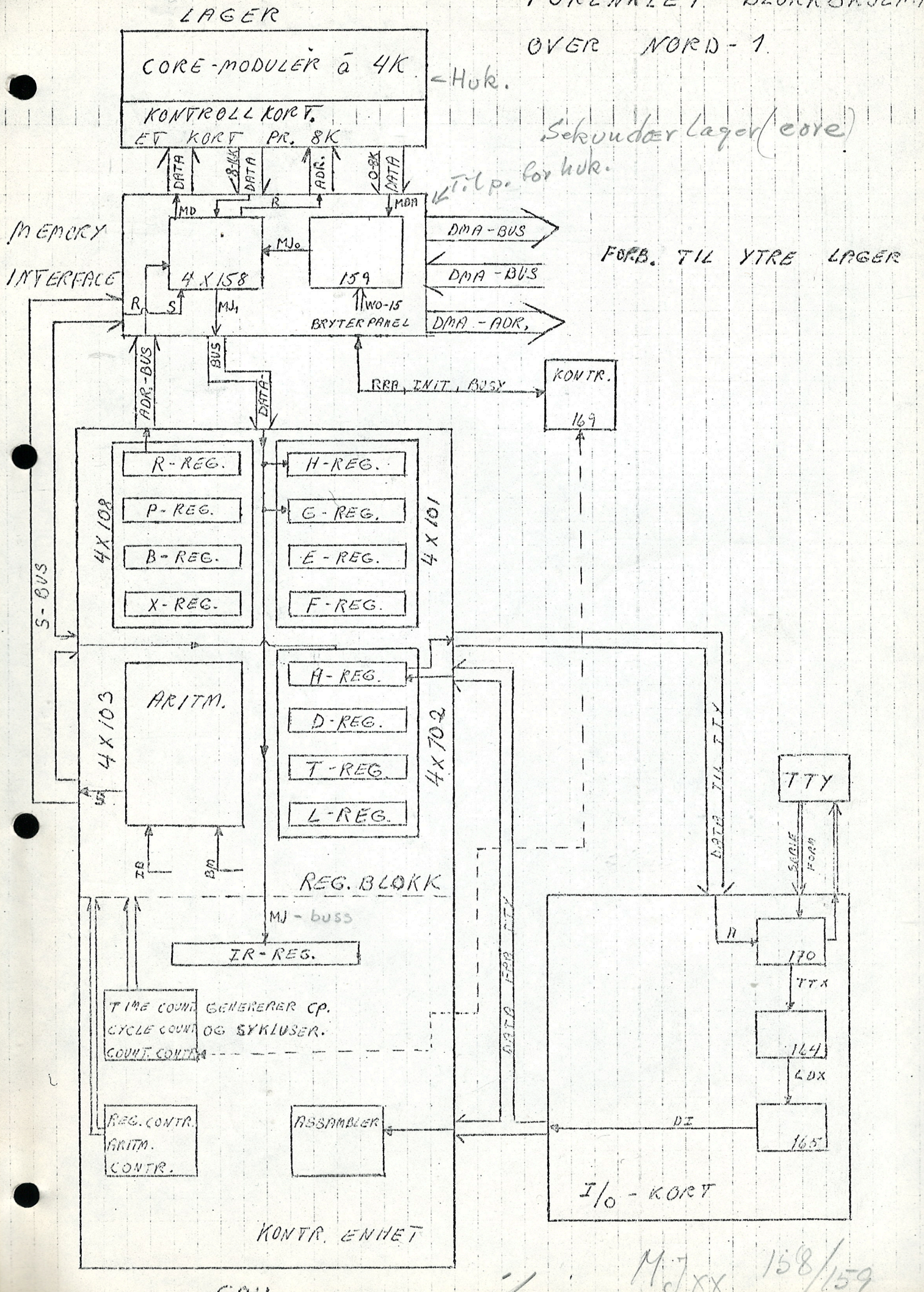

A
B
sum reg.
FA
Augendreg.
Addend reg.


MJ-BUS
FRA YTRE ENH.
DI
XB-BUS
HGEF
101
ADTL
102
IR-REG.
150
BM og $\overline{BM}$
IB-BUS
ARITM.
103
ARITM. KONTR.
115
S-BUS
DEL AV
ARITMETIKK
TIL LAGER
DEL AV
KONTROLLENHET



INTERFACE TIL LAGER.

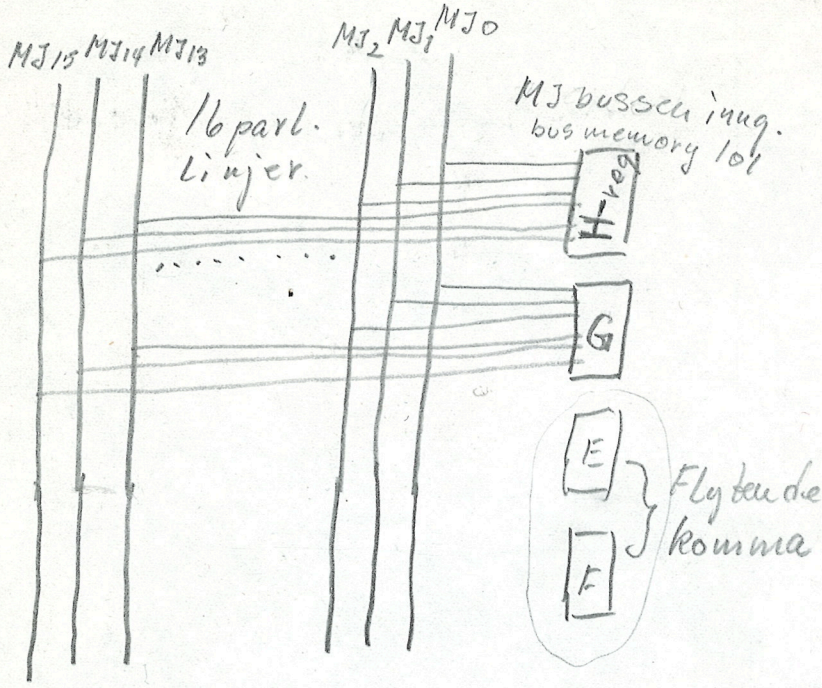
FORENKLET BLOKKSKJEMA OVER NORD-1



MJXX 158/159

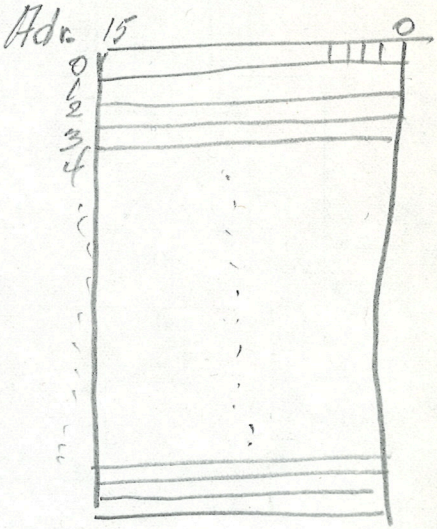
IR - Instruksjonsreg.

MJxx 158/159



IR - reg.

16 bits ord i hvk.

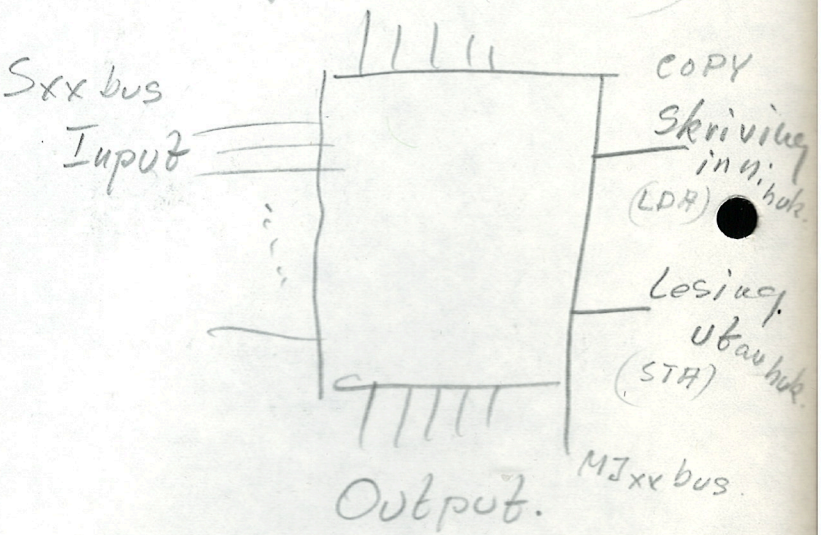


2^{10} = 1024 = 1k ord

4k = 4096

(Størrl. av hvk.)

Nrpåligene Adr. (bussen)



IR-reg.

Utg. (vipper)

IR 15 14 ------ 0

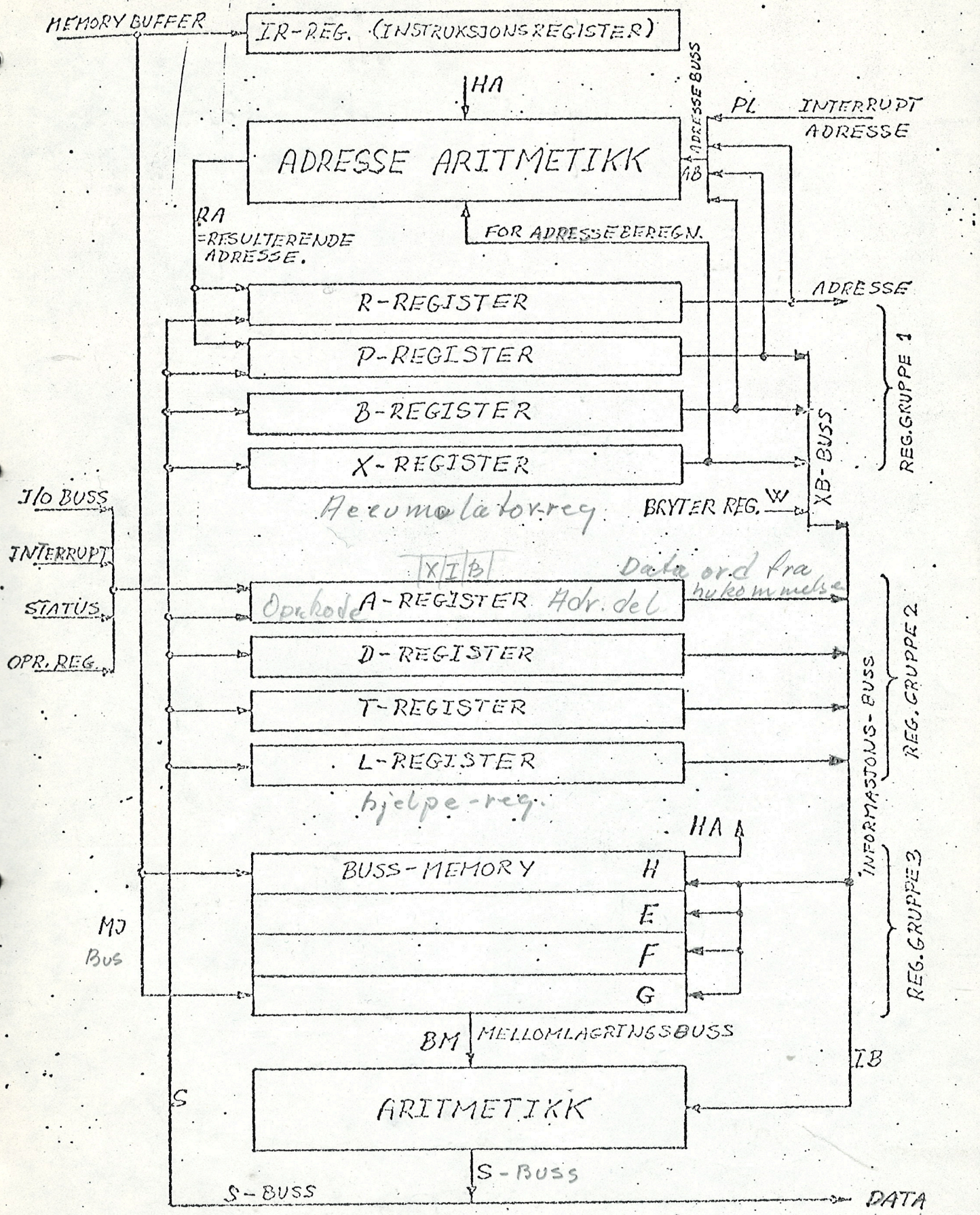


FIG. 2. GENERELLE REGISTRE OG ARITMETIKK.

Skjema 103