4) Boolean Exp.: \( \text{TT} \)

3) Draw Circuits

2) Theorems/Axioms

- \( A\overline{B} \) Complement / \( \overline{A} \overline{B} \) Pos.
- Two's Complement / Neg. Pos.
- \( 0, 1, 2, 4, 8, 16 \) Base 2, \( 0, 2, 7, 8, 16, 20 \) Base 10

1) Problems in Appendix A of Cartes

Lecture 7
• We write a program that reads an element from an array we call DATA1, then decrements the element and stores it on Array DATA2

• The program structure follows the following outline
  • Read an element of the array
  • decrement it
  • store it
  • Repeat for all elements of the array - this means we need to create a loop

• We formalize the algorithm by placing it in Pseudocode and implement the pseudocode in assembler

• Need to initialize counter, index registers and accumulators

• Write the code so that the arrays are accessed, the pointers incremented (or decremented) and test for end of the array

• Loop to repeat
  • Test for the end of the array
  • Get out of the loop after finishing the last array element
This program reads an element from the array DATA1, decrements the element and stores it on array DATA2.

**Pseudocode**

- Initialize accumulator A, index register X and counter
- Loop
  - Put element in Accumulator A
  - Decrement A
  - Store element in DATA2
  - Increment X
  - Decrement counter
- Go to Loop as long as COUNTER is not 0
- Exit the program
LDA X: (ea) → A

Same as above for X

INDEXED

Immediate "ea" is the value (x or y)

IMMEDIATE

EA = EFFECTIVE ADDRESS

X:

LDA:

ABX:

ADD B to A

ABA:

(2) + (3) → A

"ADD B to A"
BEQ (ca) - 4 PC  
BNE (ca) - 4 PC  
BRANCH (Jump) IF Z is set

STA : (A) - ce  
Store A in memory

INC : (A) + 1 - ce  
Increment A

DEC : (A) - 1 - ce  
Decrement A
ORG $C150
DATA1 FCB 1,2,3,4,5,6,7,11,22,33
DATA2 RMB 10
COUNTER RMB 1

ORG $C100
LDAA #10
STAA COUNTER
LDX #DATA1

LOOP
LDAA 0,X
DECA
STAA 10,X
INX
DEC COUNTER
BNE LOOP
SWI