

Nombre: \_\_\_\_\_

**¡Anota tu nombre y número de sección en todas las hojas del examen AHORA! (penalidad de 5 puntos)**

**Tienes 2 horas para completar todos los problemas. Lee cuidadosamente todo el examen antes de empezar a trabajar. Muestra todo el trabajo conducente a tu contestación. Podrás recibir crédito parcial por contestaciones parciales siempre y cuando muestres tu trabajo por escrito. Usa tu tiempo inteligentemente. Exitó!**

INEL 4206 Staff

Nombre: \_\_\_\_\_

1	30
2	30
3	30
4	10
<b>Total</b>	100

Nombre: \_\_\_\_\_

**Problem 1. (30 points) The Case of the Mysterious Easy I Machine Code Program**

a) **(15 points)** The following table shows the machine code version of a mysterious Easy I program. Your job is to reconstruct the assembly language version of the code and fill the corresponding column. You may refer to Appendix A for information on the Easy I ISA and Appendix C for some useful binary to decimal conversions.

Instruction Address	Assembly Code	Machine Code (binary)	Times Executed	Cycles
100	JUMPi LOOP	0 00011 0001110000		
102	I :	0 00000 0001101010		
104	SUM:	0 00000 0000000000		
106	X1 :	0 00000 0000000010		
108	X2 :	0 00000 0000000100		
110	X3 :	1 11111 1111111111		
112		1 00101 0001100110		
114		0 00010 0010000010		
116		1 00101 0001100110		
118		1 00111 0001101000		
120		0 00100 0001101000		
122		0 00101 0001100110		
124		0 00111 0000000010		
126		0 00100 0001100110		
128		0 00011 0001110000		
130	EXIT :			

} Variables

Nombre: \_\_\_\_\_

- b) **(5 points)** What is the contents of memory cell with address 104 at the end of the program segment (when PC = 130)

Value in memory address 104 =

- c) **(10 points)** Calculate the CPI achieved by the program. Fill the additional columns in the previous table. You may refer to the appendices for the number of cycles taken by each instruction to execute.

CPI =

Nombre: \_\_\_\_\_

**Problem 2. (30 points) Easy I Assembly Language Programming**

```

int a = A;
int b = B;
int res = 0;
int t = 0;
main() {
    while(b >= 0) {
        t = a;
        while (t >= 0) {
            res = res + t + b;-----
-----
            t = t - 1;
        }
        b = b -1;
    }
}

```

- (a) **(5 points)** What is the value of the *res* variable at the end of the program for the following values of *A* and *B*:

<b>A</b>	<b>B</b>	<b>res</b>
<b>1</b>	<b>1</b>	
<b>3</b>	<b>2</b>	
<b>3</b>	<b>3</b>	

- (b) **(5 points)** Express in your own words but precisely what is the mathematical relationship between *res* and *A* and *B*. You may find it handy to use the symbol  $\Sigma$  to express summation.

Nombre: \_\_\_\_\_

(c) **(20 points)** Write a segment of Easy I assembly code equivalent to the above high level language program. Use the HLL code to comment your assembly language code.

Nombre: \_\_\_\_\_

**Problema 3. (30 points) Easy I Microprocessor Implementation**

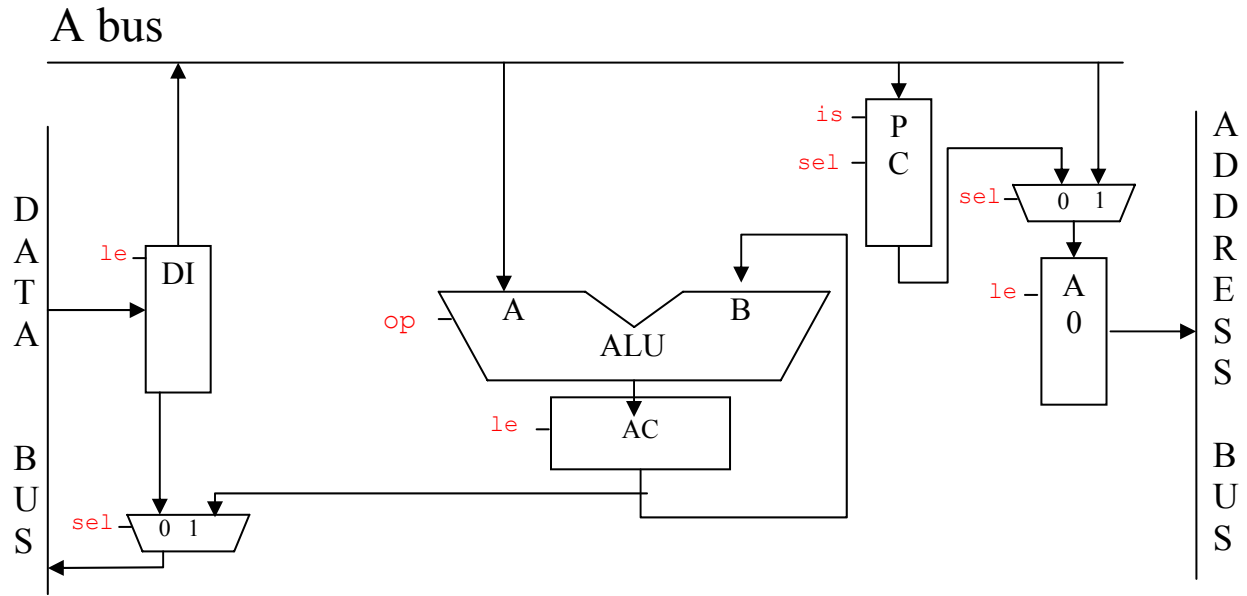
In this exercise, you will extend the instruction set of the Easy I processor discussed in class (refer to Appendix A for information about the Easy I) with a new instruction described in the following table:

Symbolic Name and Description	Assembler Example	Action
SUB	SUBi X	$AC \leftarrow X - AC$

- a) ( 10 points ) Show any changes to the Easy I datapaths necessary to implement the new instruction. Show all you changes on the diagram in the next page. Provide a concise textual description of your proposed modifications in the box below. **YOU MUST NOT MODIFY THE INTERNALS OF THE ALU.**

Nombre: \_\_\_\_\_

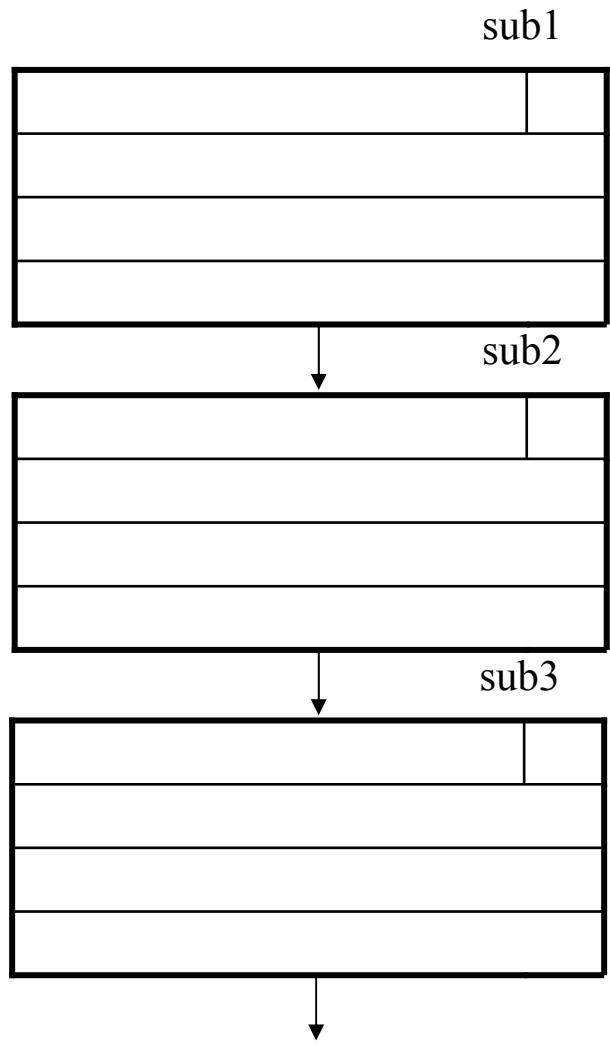
Easy I Datapaths





Nombre: \_\_\_\_\_

b) ( 10 points ) Show a level 2 flowchart implementing the SUB instruction. We provide enough room for your flowchart below, but you may not need all the states depicted below.



Nombre: \_\_\_\_\_

- c) ( 10 points ) Show any modifications to the Easy I state transition table described in class. Show your changes on the table in the following page. Also, provide a brief textual description of your changes in the box below.



Nombre: \_\_\_\_\_

Easy I State Transition Table

Curr State	opcode	I	AC:15	Next State	ALU op	Mem OP	PC sel	PC is	DI le	AC le	AO sel	AO le	EDB sel
reset1	xx xxx	x	x	reset2	XXX	NOP	01	X	0	0	X	0	X
reset2	xx xxx	x	x	fetch	XXX	NOP	10	1	0	0	0	1	X
fetch	00 00x	0	x	sopr	XXX	NOP	11	X	1	0	X	0	X
fetch	00 010	0	x	brn1	XXX	RD	11	X	1	0	X	0	X
fetch	00 011	0	x	jump	XXX	RD	11	X	1	0	X	0	X
fetch	00 100	0	x	store1	XXX	RD	11	X	1	0	X	0	X
fetch	00 101	0	x	load1	XXX	RD	11	X	1	0	X	0	X
fetch	00 11x	0	x	aopr	XXX	RD	11	X	1	0	X	0	X
aopr	00 110	x	x	fetch	AND	NOP	10	1	0	1	0	1	X
aopr	00 111	x	x	fetch	ADD	NOP	10	1	0	1	0	1	X
sopr	00 000	x	x	fetch	NOTB	NOP	10	1	0	1	0	1	X
sopr	00 001	x	x	fetch	SHRB	NOP	10	1	0	1	0	1	X
store1	xx xxx	x	x	store2	XXX	NOP	11	X	0	0	1	1	X
store2	xx xxx	x	x	fetch	XXX	WR	10	1	0	0	0	1	1
load1	xx xxx	x	x	load2	XXX	NOP	11	X	0	0	1	1	X
load2	xx xxx	x	x	load3	XXX	RD	11	X	1	0	X	0	X
load3	xx xxx	x	x	fetch	A	NOP	10	1	0	1	0	1	X
brn1	xx xxx	x	0	fetch	XXX	NOP	10	1	0	0	0	1	X
brn1	xx xxx	x	1	brn2	XXX	NOP	10	1	0	0	0	1	X
brn2	xx xxx	x	x	fetch	XXX	NOP	10	0	0	0	1	1	X
jump	xx xxx	x	x	fetch	XXX	NOP	10	0	0	0	1	1	X

**Problema 4 ( 10 points) Evaluación del curso**

**(a) Menciona los tres aspectos que mas te gustan de la clase INEL 4206 en orden decreciente de importancia.**

**a.**

**b.**

**c.**

**(b) Menciona los tres aspectos que menos te gustan de la clase INEL 4206 en orden decreciente de importancia.**

**a.**

**b.**

**c.**

**(c) En una escala de 0 (no mejoría) al 5 (mejoría excelente) como evaluarías la respuesta del professor a los comentarios que sugeriste durante el examen parcial 1. Explica brevemente.**

**1**

**2**

**3**

**4**

**5**



## Appendix C. Additional Information

Decimal	Binary
112	0001110000
106	0001101010
0	0000000000
2	0000000010
4	0000000100
-1	1111111111
130	0010000010
104	0001101000
102	0001100110

Instruction	Cycles
COMP	2
ShR	2
BrN	3 if branch taken 2 if not taken
JUMP	2
STORE	3
LOAD	4
AND	2
ADD	2