**Kettering University**

Microcomputers I

**Class Exercise Packet III**

Spring 2022

Name:

**Notes**

* Answer the questions provided in this handout during normal class times when the instructor asks you to do so, and then upload it to Bb (to get extra credit) using one of the following methods when you are notified:
* You may directly answer the questions in the electronic copy of the handout posted on Bb, and then convert it to .pdf format to submit the packet.
* You may answer the questions in the paper copy, and then scan and convert it to .pdf format to submit the packet.
* (Preferred) You may answer the questions in the paper copy, then transfer your answers to the electronic copy, and convert it to .pdf format to submit the packet.
* In multiple-choice questions, highlight the correct choice(s).
* Exercises are **open** book/notes unless specified otherwise.
* If you know the answers, please **teach** other students; otherwise, **learn** from other students, but do NOT copy from them!
* If you **run out of time** on a question, answer that question later but **before** you walk into the classroom next time.
* **Do NOT hesitate to stop by my office if you need help!**
* This handout (and not anything else!) in .pdf may be collected for grading purposes electronically **anytime**. So, keep it updated!

1. Execute the following code manually and determine the contents of the registers as they change. Use the diagram below to show how the stack grows and shrinks:

4000

lds #$4000

ldd #$9876

ldy #$2345

pshy

pulb ; B <= …

pula ; A <= …

4001

pshd

pshb

puly ; Y <= …

1. Configure odd pins of PORT B as output, and the even pins as input:

Memory

1. Configure PA2 as output, leave the rest of the pins as they are:
2. Send the value $34 to Port P (outside world):
3. What should be displayed on the following 7-segment displays? **Note**: a is the LS segment.

PB7

PB6

PB5

PB4

PB3

PB2

PB1

PB0

$5B

0

1

0

0

0

0

0

b

c

d

e

f

g

h

a

PP0

PB7

PB6

PB5

PB4

PB3

PB2

PB1

PB0

$65

1

0

1

1

PP0

1. Write a piece of assembly code to display digit 9 on the second display from the right:
2. Write the 4 x 7-segments Demultiplexing Algorithm legibly and neatly: (This question is closed notes!)

PB7

PB6

PB5

PB4

PB3

PB2

PB1

PB0

PP0

PP1

PP3

PP2

b

c

d

e

f

g

h

a

1. Using IO poling, draw a neat and legible flowchart to toggle LED0 when the pushbutton is *released*.

**Note**:

In general, some blocks of the flowchart may be combined when translating into HCS12 language, and some blocks may be translated into 1+ instructions.

PB0

HCS12

PJ1

PH3

GND

LED0

Vcc

PH3

1. How many interrupt vectors are there for the 8 pins of Port H?
2. If a Port H pin interrupts the CPU, how do you know which pin it was?
3. Model the RTI period algebraically:

0

0

0

1. Draw a flowchart for the following C program (from lecture slides):

// Infinite loop

while (-1)

{

for (**i** = 0; **i** < 10; i++) // i is lookup-table index of first digit in next pattern

{

for (j = 0; j < 100; j++) // duration of each pattern (there are 6 digits in a pattern)

{

for (**k** = 0; **k** < 6; k++) // starting with index i, display 6 digits in current pattern

{

PTP = 0xFF; // turn displays off

PORTB = SegPat[i+k]; // send next digit

PTP = digit[k]; // turn on next display

for (m = 0; m < 2000; m++); // display remains on in this wait loop

}

}

}

} // While