Landon Leigh ITSC 3181 HW 3

1. 1. 16 = 10000 -> 2’s compliment = 010000

9 = 1001 -> 2’s = 001001

010000 + 001001 = 011001 no overflow

* 1. 27 = 11011 -> 2’s = 011011

31 = 11111 -> 2’s = 011111

011011 + 011111 = 111010 no overflow

* 1. -4 = 100 -> 000100 -> 111011 + 1 = 111100

19 = 10011 -> 010011

111100 + 010011 = 1001111 yes overflow so 001111

* 1. 3 = 11 -> 000011

-32 = 100000 -> 011111 + 1 = 100000

000011 + 100000 = 100011 no overflow

* 1. -16 = 10000 -> 01111 + 1 -> 10000

-9 = 1001 -> 0110 + 1 -> 110111

10000 + 110111 = 1100111 yes overflow so 100111

* 1. -27 = 11011 -> 100100 + 1 = 100101

-32 = 100000 -> 011111 + 1 = 100000

100101 + 100000 = 1000101 yes overflow so 000101

1. Main:

li v0, 5

syscall

move t0, v0

li v0, 5

syscall

mov t1, v0

loop:

beq t1, zero, exit

add t2, t2, t0

addi t1, t1, -1

b loop

exit:

li v0, 1

mov a0, t2

syscall

1. Implementation 1: 1(1 + 1 + 1) + 3(5 + 9 + 2 + 2 + 5) + 1(5) = 92 cycles

Implementation 2: 1(1 + 1 + 1) + 3(9 + 5 + 2 + 2 + 5) = 72 cycles

So, implementation 2 is faster.

1. main:

mov ax, data

mov ad, ax

lea si, num1

lea di, num2

lea bx, result

mov cx 5

loop:

mov al, [si]

add al, [di]

mov [bx], al

inc bx

inc si

inc di

loop

mov ah, 4ch

int 21h

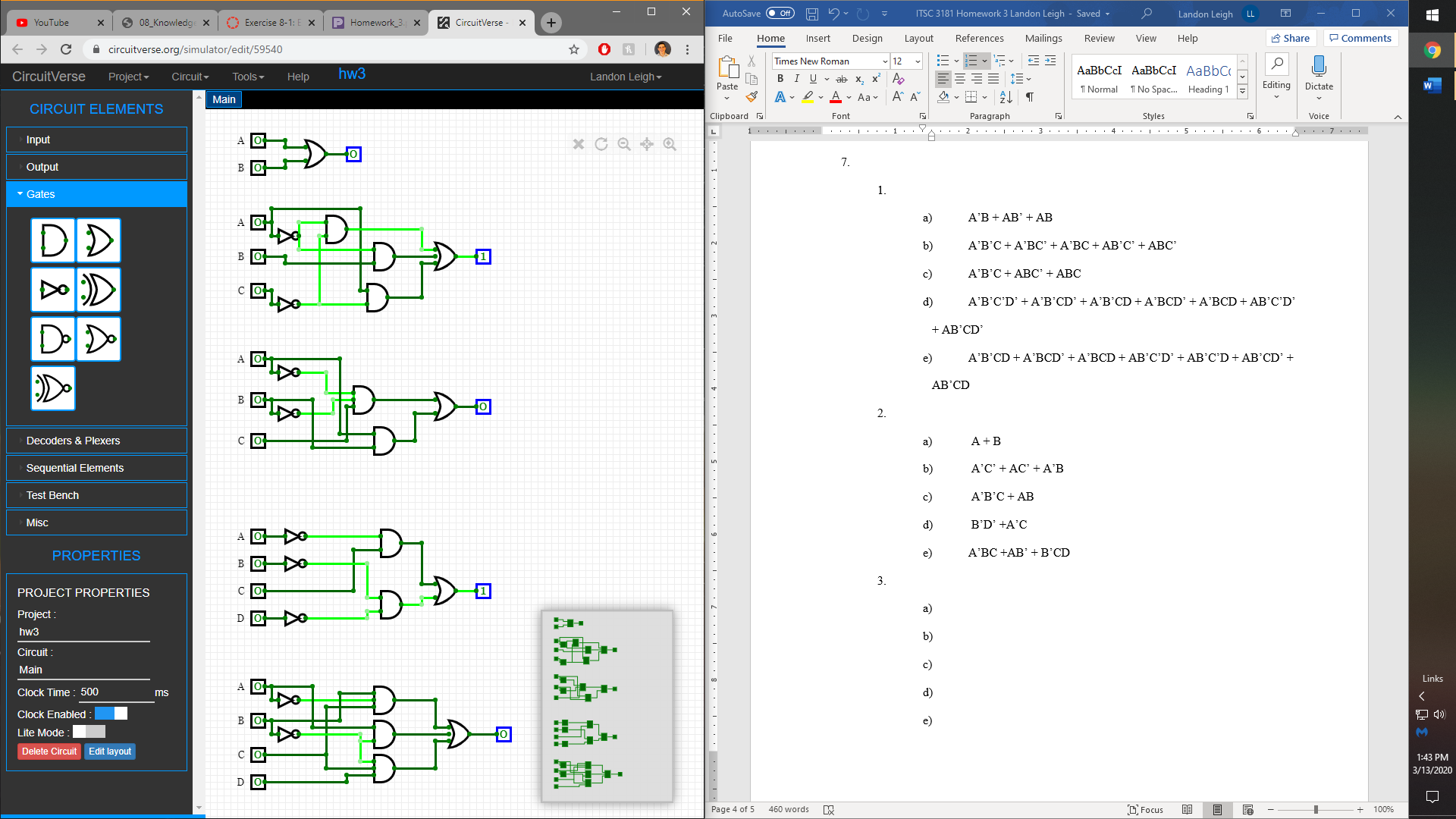
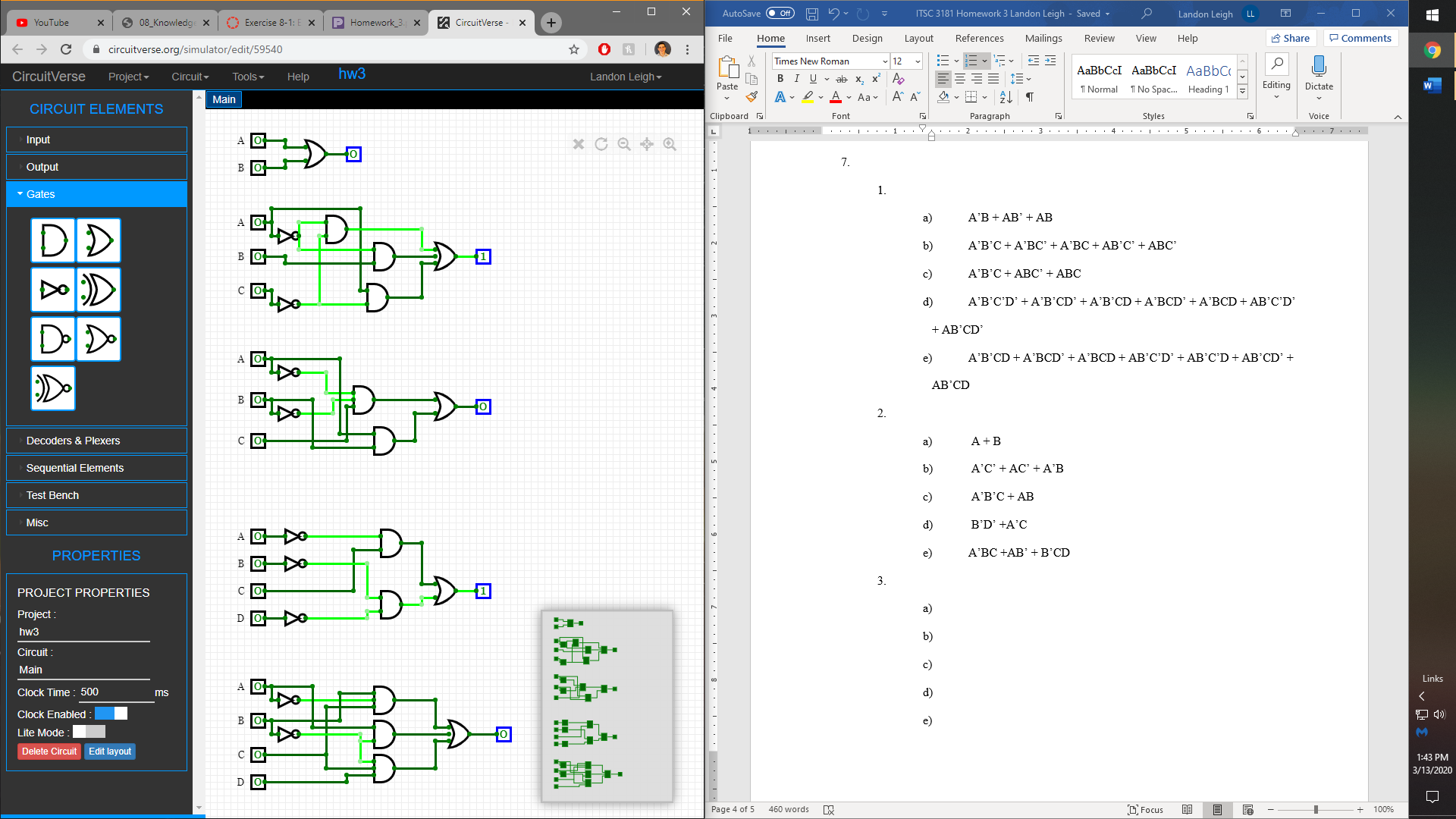
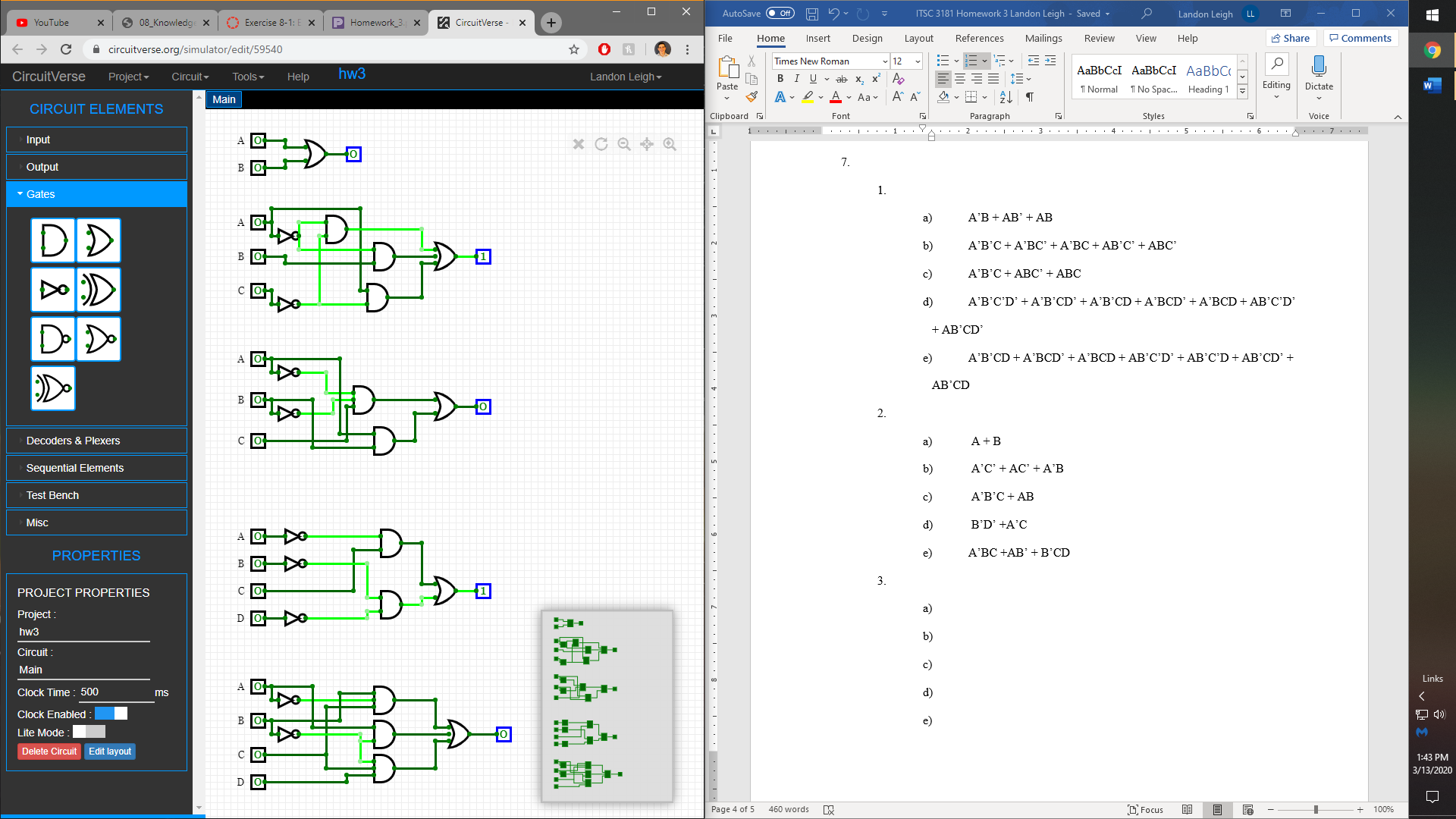
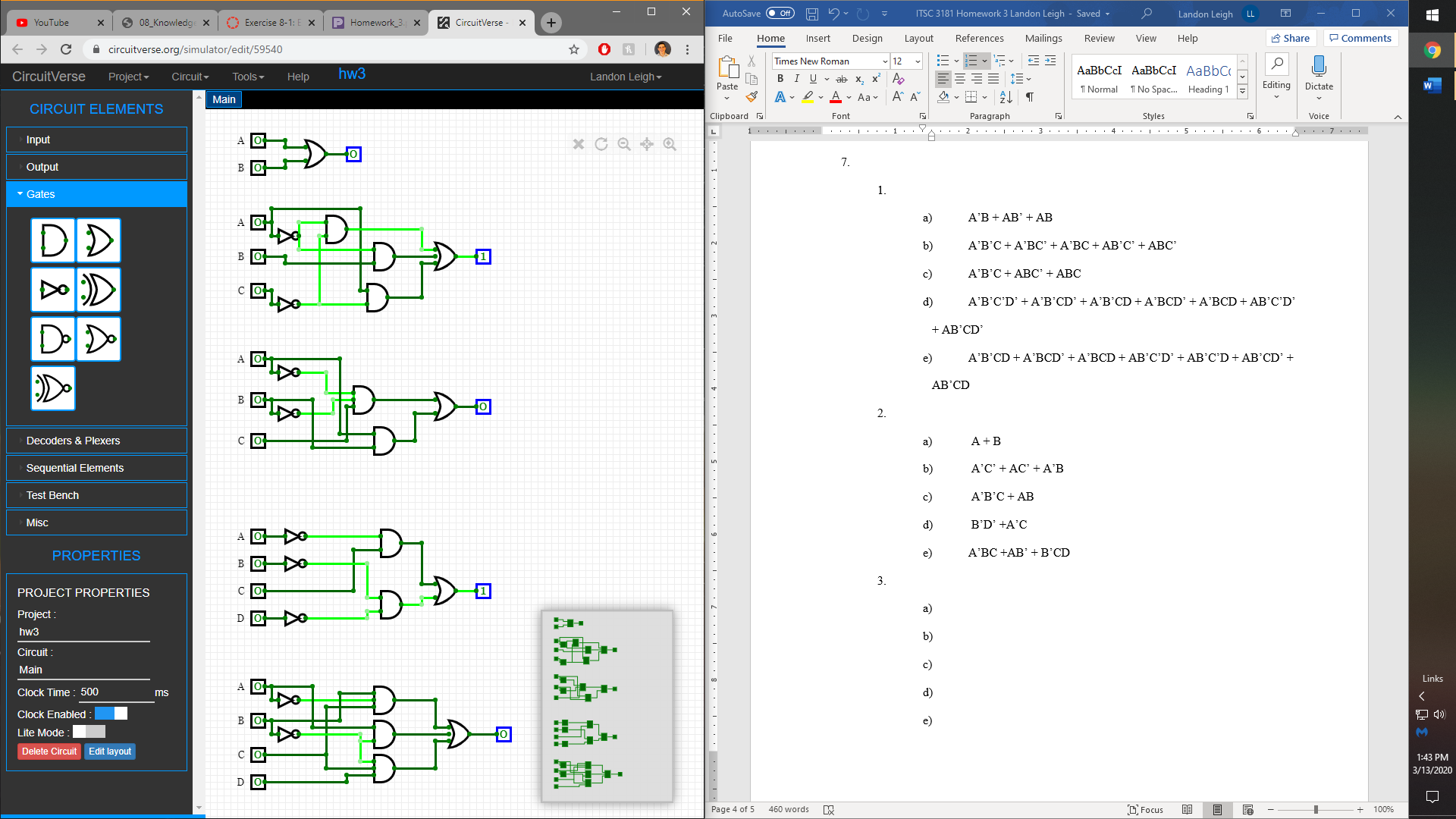


|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Instruction Class | CPI | # inst \* 10^6 | total cyc \* 10^6 | Cycles % |
| add | 1 | 2\*10^6 + 1 | 2\*10^6 + 1 = 2000001 | 7.14 |
| mul | 20 | 1\*10^6 | 20\*10^6 = 20000000 | 71.428 |
| Load/store | 2 | 2\*10^6 + 1 | 4\*10^6+2 = 4000002 | 14.28 |
| branch | 2 | 1\*10^6 + 1 | 2\*10^6 + 2 = 2000002 | 7.14 |

Total cycles = 28000005

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Instruction Class | CPI | # inst \* 10^6 | total cyc \* 10^6 | Cycles % |
| add | 1 | 2001000 | 2001000 | 7.14 |
| mul | 20 | 1000000 | 20000000 | 71.145 |
| Load/store | 2 | 2001000 | 4002000 | 14.29 |
| branch | 2 | 1001000 | 2002000 | 7.14 |

Total cycles = 28005000

1. * 1. A’B + AB’ + AB
     2. A’B’C + A’BC’ + A’BC + AB’C’ + ABC’
     3. A’B’C + ABC’ + ABC
     4. A’B’C’D’ + A’B’CD’ + A’B’CD + A’BCD’ + A’BCD + AB’C’D’ + AB’CD’
     5. A’B’CD + A’BCD’ + A’BCD + AB’C’D’ + AB’C’D + AB’CD’ + AB’CD
   1. 1. A + B
      2. A’C’ + AC’ + A’B
      3. A’B’C + AB
      4. B’D’ +A’C
      5. A’BC +AB’ + B’CD
   2. 1. 
      2. 
      3. 
      4. 
      5. 