Green highlight= go forward a step

Blue highlight=go back 1 or more steps

**Possible Plan for Approaching Assignment #6**

Start with the first line of data

Step 1

* Look for the addresses 0x40000810 or 0x40000C18

Step 2

* if address = 0x40000810
  + it is the S-to-D command
  + save that information(possibly as a Boolean)
* if address = 0x40000C18
  + it is the D-to-S command
  + save that information(possibly as a Boolean)
* read and store the information in the cycle column for this line(“Wr” or “Rd”)
  + (“wr”=write and “rd”=read)
* store the Rel Time of the given command (tricky)
  + to save the correct rel time go to the next line and store that rel time value
  + save it in the appropriate group
    - “Read S-to-D”
    - “Read D-to-S”
    - “Write S-to-D”
    - “Write D-to-S”.
  + this information will be used at the end of the program
* read and store the information in the data column for this line
  + if Data=0 (0 words)
    - print the correct output (“line #: [write or read] [“S-to-D or D-to-S”][number of words])
    - go back to step 1
  + if Data > 0
    - print the correct output (“line #: [write or read] [“S-to-D or D-to-S”][number of words])
    - proceed to step 3

Step 3

* Search the correct range for the next lines you will read in

(0x40000818 to 0x40000C14 contains the data of the S-to-D command)

(0x40000C20 to 0x4000101C contains the data of the D-to-S command)

* note: this range will vary based on the number of bytes of data(the data we found in step 2)
  + for example
    - We are given a S-to-D command
    - data= 0000000C = 12 bytes = 6 words
    - typically for an S-to-D command we look in the range: 0x40000818 to 0x40000C14
    - Since there are 12 bytes of data, start from 0x40000818 and move up 12 bytes(6 words) to 0x40000824
    - so you would look in the range 0x40000818 to 0x40000824 for that S-to-D command
* if at any time address = 0x40000810 or 0x40000C18
  + go back to step 1
* once the correct range is found
  + proceed to step 4

Step 4

* in a loop collect all the values(lines) in the correct range
  + this loop will be broken if/when:
    - find a new S-to-D or D-to-S command ( address = 0x40000810 or 0x40000C18

OR

* + - you have all the information for all the words you are looking for
* once you have all the values stored
  + compare the 1st stored address with the 2nd stored address
* if the 1st address < 2nd address
  + it’s in ascending order
* if the 1st address > 2nd address
  + it’s in descending order
* for example:
  + address1=40000818
  + addresss2=40000820
  + address1 < address2
  + this is ascending order
* another example:
  + address1=4000082c
  + addresss2=40000820
  + address1 > address2
  + this is descending order
* once you find out if its ascending or descending save that information(possibly as a Boolean)

Step 5

* now go through each line in the loop
* take the data in each line and separate it accordingly
  + for example:
    - “1A022DFE” becomes:
    - word 0: “1A02”
    - word 1: “2DFE”
* translate the words you just parsed into bits
* find the value for the description of the word by checking the correct bits(find this in the table below)
  + for example:
    - word 0=”1A02”= 0001 1010 0000 0010
    - look at bits 14-13(as instructed by the table)
    - bit 14-13=00=0
    - so for word 0 description is “no recording”
* save the description for this word
* find and save the correct field name based on the word number (find this in the table below)
* print the correct output
  + (“line #, word #, field name, description)
* repeat this process until you go through every line in the loop
* once you finish the loop
  + if there are more lines after
    - go back to step 1
  + if not
    - go to step 6

Step 6

* calculate the data rate for each type of(from the data of rel time saved in step 1)
  + “Read S-to-D”
  + “Read D-to-S”
  + “Write S-to-D”
  + “Write D-to-S”.
* The data rates should be accurate to 2 decimal places
* use Kilobits/sec or Megabits/sec or Gigabits/sec or Terabits/sec as appropriate.
* Print the data rates at the end of the output file.
  + Example.
    - Read S-to-D: 25.73 Kilobits/sec
    - Read D-to-S: 53.01 Megabits/sec
    - Write S-to-D: 154.24 Gigabits/sec
    - Write S-to-D: 1.22 Terabits/sec e.
* End program