CSARCH2 (S12) December 17, 2020
Simulation Project

Case Project: CSARCH simulator

*Develop a stand-alone executable java program [jar file] with GUI simulator

Submission type:	Submission of source code and jar file via CANVAS
Note:	self-executable jar file will be used to check your project; no self-executable java file (jar) automatic 0 for the project
Note #2:	Provide a readme file containing user's manual on how to use your apps (send as comment attachment)
Deadline:	please refer to Canvas for deadline
Demo:	via zoom (if needed)

Topic (see description below)	Group #
Decimal-32	10
Decimal-64	12
Binary-32	7
Binary-64	5
Cache simulator (BSA/LRU)	2
Cache simulator (BSA/MRU)	8
Cache simulator (FA/LRU)	1
Cache simulator (FA/MRU)	4
Cache simulator (Direct)	3
Sequential circuit binary multiplier	6
Restoring division	9
Non-restoring division	11

* IEEE-754 Decimal-32 floating point converter (including all special cases)
    - Input: Decimal and base-10 (i.e., $127.0x10^5$) – should handle more than 7 digits properly and NaN
    - Output: binary output with space as well as its hexadecimal equivalent (with option to paste result in notepad)
* IEEE-754 Decimal-64 floating point converter (including all special cases)
    - Input: Decimal and base-10 (i.e., $127.0x10^5$) – should handle more than 7 digits properly and NaN
    - Output: binary output with space as well as its hexadecimal equivalent (with option to paste result in notepad)
* IEEE-754 Binary-32 floating point converter (including all special cases)
    - Input: binary mantissa and base-2 (i.e., $101.01x2^5$) and NaN
    - Output: binary output with space as well as well as its hexadecimal equivalent (with option to paste result in notepad)
* IEEE-754 Binary-64 floating point converter (including all special cases)
    - Input: binary mantissa and base-2 (i.e., $101.01x2^5$) and NaN
    - Output: binary output with space as well as well as its hexadecimal equivalent (with option to paste result in notepad)
* Cache simulator (Block-set-associative / LRU)
    - Input: Block size, set size, MM memory size (either blocks or words), cache memory size (either blocks or words), program flow to be simulated and other parameters deemed needed (example: can simulate cache problem set # 4,5,6)
    - Output: number of cache hits, number of cache miss, miss penalty, average memory access time, total memory access time, snapshot of the cache memory. With option to output result in text file.
* Cache simulator (Block-set-associative / MRU)
    - Input: Block size, set size, MM memory size (either blocks or words), cache memory size (either blocks or words), program flow to be simulated n and other parameters deemed needed(example: can simulate cache problem set # 4,5,6)
    - Output: number of cache hits, number of cache miss, miss penalty, average memory access time, total memory access time, snapshot of the cache memory. With option to output result in text file.

* Cache simulator (Full associative / LRU)
    - Input: Block size, MM memory size (either blocks or words), cache memory size (either blocks or words), program flow to be simulated and other parameters deemed needed (example: can simulate cache problem set # 4,5,6)
    - Output: number of cache hits, number of cache miss, miss penalty, average memory access time, total memory access time, snapshot of the cache memory. With option to output result in text file.

* Cache simulator (Full associative / MRU)
    - Input: Block size, MM memory size (either blocks or words), cache memory size (either blocks or words), program flow to be simulated and other parameters deemed needed (example: can simulate cache problem set # 4,5,6)
    - Output: number of cache hits, number of cache miss, miss penalty, average memory access time, total memory access time, snapshot of the cache memory. With option to output result in text file.

* Cache simulator (Direct)
    - Input: Block size, MM memory size (either blocks or words), cache memory size (either blocks or words), program flow to be simulated and other parameters deemed needed (example: can simulate cache problem set # 4,5,6)
    - Output: number of cache hits, number of cache miss, miss penalty, average memory access time, total memory access time, snapshot of the cache memory. With option to output result in text file.

* Sequential Circuit Binary Multiplier simulator (accepts binary input. Use minimum number of bits based on the input. Max up to 16-bit; Option to show either step-by-step or “all” mode; show A and Q output after every step) with option to output result in text file.

* Non-Restoring Unsigned division simulator (accepts binary input, use minimum number of bits based on the input. Max input up to 16-bit; show either step-by-step or “all” mode; show A and Q output after every step) with option to output result in text file.

* Restoring Unsigned division simulator (accepts binary input, use minimum number of bits based on the input. Max input up to 16-bit; Option to show either step-by-step or “all” mode; show A and Q output after every step) with option to output result in text file.