

## CSARCH2

### Simulation Project

#### Case Project: CSARCH simulator

\*Develop a simulator using either

- HDL-based
- Web-based with Graphics User Interface (GUI)
- Java-based with Graphics User Interface (GUI). Submission should be in standalone executable jar file (i.e. no need to use the external java compiler). It will be considered as no submission if jar file is not executable

Submission type: HDL file, jar file or URL link of your web-based simulator and submit through Canvas

Note: 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-to-group assignment: There will be a separate file for the topic assignment

\*\*\*Simulator project description\*\*\*

\* IEEE-754 Decimal-32 floating point converter (including all special cases)

- Input: Decimal and base-10 (i.e.,  $127.0 \times 10^5$ ) – should be able to handle more than 7 digits properly (provide option for user to choose round-off method) and NaN
- Output: (1) binary output with space between section (2) its hexadecimal equivalent (3) 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.,  $127.0 \times 10^5$ ) – should handle more than 16 digits properly (provide options for user to choose round-off method) and NaN
- Output: (1) binary output with space between section (2) its hexadecimal equivalent (3) 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.,  $101.01 \times 2^5$ ), decimal mantissa and base-10 (i.e.,  $15.75 \times 10^5$ ) and NaN
- Output: (1) binary output with space between section (2) its hexadecimal equivalent (3) 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.,  $101.01 \times 2^5$ ), decimal mantissa and base-10 (i.e.,  $15.75 \times 10^5$ ) and NaN
- Output: (1) binary output with space between section (2) its hexadecimal equivalent (3) with option to paste result in notepad

\* IEEE-754 Binary-32 floating point translator (including all special cases)

- Input: 8-digit hex input or 32-bit binary input (provide separator for various sections of the input)
- Output: (1) Decimal (provide option for user to choose between fixed or floating point)  
(2) with option to paste result in notepad

\* IEEE-754 Binary-64 floating point translator (including all special cases)

- Input: 16-digit hex input or 64-bit binary input (provide separator for various sections of the input)
- Output: (1) Decimal (provide option for user to choose between fixed or floating point)  
(2) with option to paste result in notepad

\* IEEE-754 Decimal-32 floating point translator (including all special cases)

- Input: 8-digit hex input or 32-bit binary input (provide separator for various sections of the input)
- Output: (1) Decimal (provide option for user to choose between fixed or floating point)  
(2) with option to paste result in notepad

\* IEEE-754 Decimal-64 floating point translator (including all special cases)

- Input: 16-digit hex input or 64-bit binary input (provide separator for various sections of the input)
- Output: (1) Decimal (provide option for user to choose between fixed or floating point)  
(2) with option to paste result in notepad

\* Binary multiplication simulator

- Input: binary or decimal (max up to 16-bit)
- Output: (1) pencil-and-paper (2) Booth's (3) Extended booth's algorithm
  - option to show step-by-step or "all" mode of the intermediate result
  - with option to output result in text file.

\* Sequential Circuit Binary Multiplier simulator

- Input: binary or decimal (use minimum number of bits; max up to 16-bit)
- Output:
  - Option to show either step-by-step or "all" mode of the A and Q output after every step
  - option to output result in text file.

\* Non-Restoring Unsigned division simulator

- Input: binary or decimal (use minimum number of bits; max up to 16-bit)
- Output:
  - Option to show either step-by-step or “all” mode of the A and Q output after every step
  - option to output result in text file.

\* Restoring Unsigned division simulator

- Input: binary or decimal (use minimum number of bits; max up to 16-bit)
- Output:
  - Option to show either step-by-step or “all” mode of the A and Q output after every step
  - option to output result in text file.

\*BCD generator and translator.

- For BCD generator:
  - Input: decimal; output: unpacked BCD, packed BCD, densely-packed BCD;
- For Densely-packed BCD translator:
  - Input: Densely-packed BCD, output: decimal
- Note: output with option to output result in text file

\*UNICODE

- Input: Unicode (with invalid Unicode check)
- Output: UTF-8; UTF-16; UTF-32 [format: xx xx xx ; where x is hex nibble]
- Note: output with option to paste result in notepad