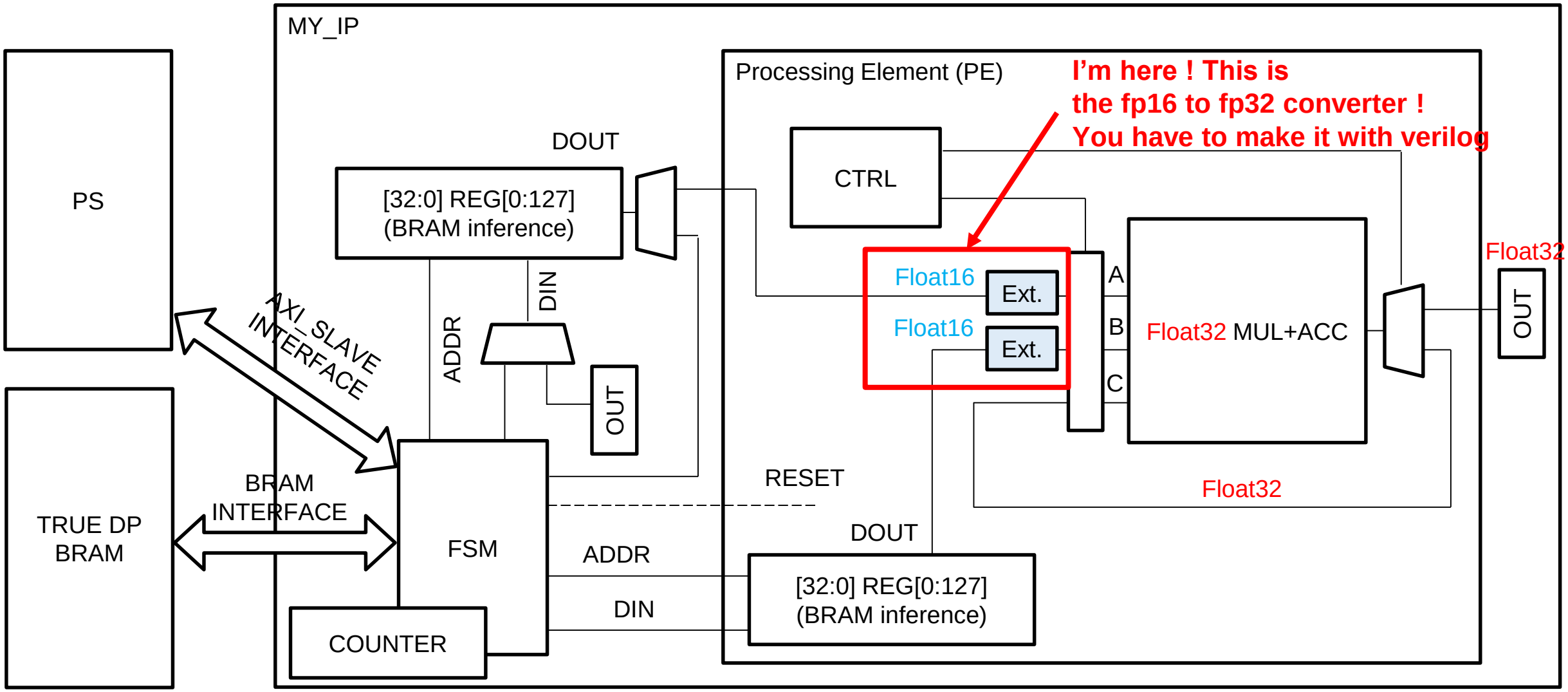
Term Project #1: FAQ and supplementary slides for Term Project #1

05/18/2018

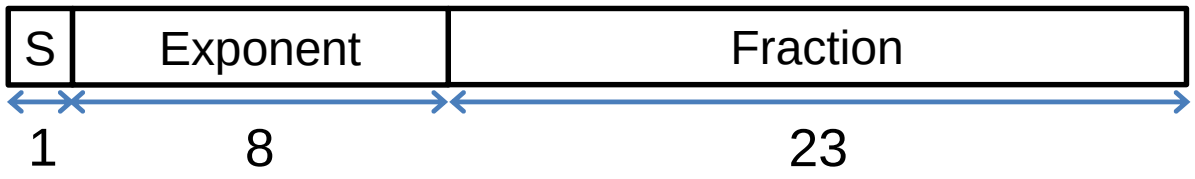
4190.309A: Hardware System Design
(Spring 2018)

Q1: Overview: Matrix-vector multiplication IP

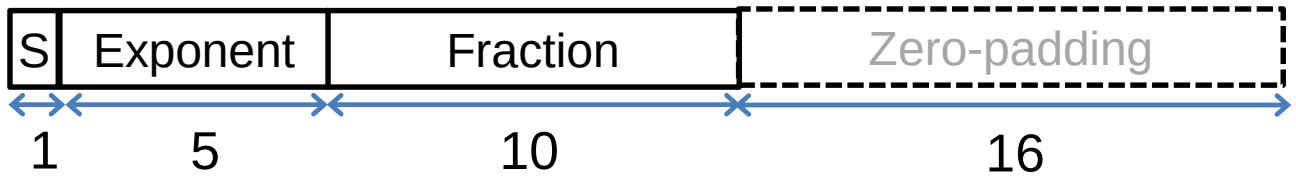


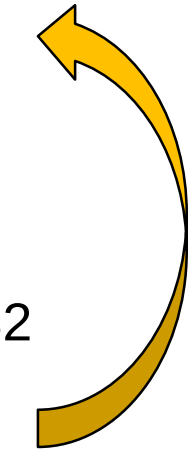
Q1: Data format used in Term Project #1

- Data format
  - **FLOAT32**: IEEE754 single-precision (**used for computation**)
    - 32bit (1bit sign / 8bit exponent / 23bit fraction)



  - **FLOAT16**: IEEE754 half-precision (**used for BRAM**)
    - 16bit (1bit sign / 5bit exponent / 10bit fraction)
    - You'll use this format in SW, but have to convert it into FLOAT32 right before computing with floating-point MAC IP





You must implement FLOAT16 to FLOAT32 converter for internal use in MV multiplication IP

Q1: Data format used in Term Project #1

- Format conversion with **hardware** !
  - Computers internally represent floating point number using binary number format in 1.xxx * 2^{yyy} formats. Here is a very good material that describes floating-point format details [Floating-point Guide]
  - Note that "**only numbers in normalized form**" will be used for test & scoring
  - Normalized form in Float16: [1b Sign] * 1.[10b Fraction] * $2^{[5b\ Exponent] - 15}$
  - Normalized form in Float32: [1b Sign] * 1.[23b Fraction] * $2^{[8b\ Exponent] - 127}$
  - Here is an high-level description of what you should do with conversion logic
    - (1) You need to implement a simple extender for fraction (from 10b to 23b)
    - (2) You need to adjust bias from 15 to 127 for exponent (with extension from 5b to 8b)
  - Hints: Details are already described in C, check func. *f16_to_f32* in *zync.cpp*

[Floating-point Guide] http://floating-point-gui.de

Q1: Data format used in Term Project #1

- Other references
  - [Wikipedia] https://en.wikipedia.org/wiki/Single-precision_floating-point_format
  - [Floating-point Guide] http://floating-point-gui.de
  - [A Tutorial on Data Representation from NTU] http://www3.ntu.edu.sg/home/ehchua/programming/java/datarepresentation.html

Other Questions (Updated at 5/18)

- FAQ
  - Q: What modules I have to modify?
    - A: You can freely modify myip modules, but it'd be pe_controller.v mainly. Also, you should write down the corresponding software for data movement, execution, and result write-back to DRAM (zynq.cpp)
  - Q: What is hls folder?
    - A: It's external library for half-precision floating points. In fact, you don't have to care about it. Also, hls_half.h, hls_fpo.h is just used for this project.