Keegan Smith

ADS

HW3: Pipeline Design

10/18/23

1. For our implementation, each pipeline stage needs to perform some multiplications, additions, as well as a comparison. Multiplication is performed over the real and imaginary parts of z, and addition is performed with the result of the multiplication as well as the real and imaginary portions of c. We wish to group our signals into records for ease of handling. Construct a record whose elements are two ads\_complex types, one for z and one for c.

A screen shot of a computer program

Description automatically generated

1. Declare an entity that has a reset signal and clock signal of type std\_logic and uses the previously constructed record as type for a stage\_input signal and a stage\_output signal.

A screen shot of a computer program

Description automatically generated

1. A computer screen shot of a program code

   Description automatically generatedExtend the entity to include a generic parameter for an arbitrary threshold of type ads\_sfixed, as well as a stage number of type natural.
2. A screen shot of a computer code

   Description automatically generatedAdd an entry of type natural called stage\_data and an entry of type boolean called stage\_overflow to the record type from Question 1.
3. Construct the architecture of the stage reducing the number of multiplications necessary. Drive your output according to the following rules:

* The *z* output should be the value after computing *z^2 + c.*
* The *c* output should be the same as the c value which was given to the entity.
* The *stage\_overflow* output should be true if it was true on the *stage\_overflow* input, or if it went over the threshold on this stage.
* The *stage\_data* output should be the value on the *stage\_data* input if the *stage\_overflow* input is true, else it should be the current stage number.

A screen shot of a computer program

Description automatically generated