Architecture code updates LOG

4/9:

**Arc\_spec**: made (all?) necessary changes. Took out ultrascale+, replaced with 7 series requirements

**Bit\_locator** : going to take some work. Instead of rows being just incremented 0 at bottom and then going until top, is divided into 2 halves and then split off from the center being row 0 for each half and then reaching for the ends.

 Need to decide convention for get min/max row. Am going to say max and number of rows is therefore number of ROWS PER HALF. This affects other python files like frame.py.

HOWEVER, To figure out what SLR are in, old code gets the bel\_y and then see if within the min/max y.

ASSUMING SLRs are ABOVE HALF-LEVEL DIVISION (so have a center/zero line for rows within each SLR)

Need to update dependencies

ASSUMING BEL\_X AND BEL\_Y ARE ABSOLUTE

# ASSUMING get\_min\_clock\_region\_row\_idx (and the max version) ARE ABSOLUTE ROWS, OR CANT TELL BETWEEN SLRs

#ASSUMING get\_min\_row\_majors (and the max version) are ABSOLUTE

**frame.py**: started, but not finished.

4/14:

Changed bit to differentiate halves from ‘top\_bit’ to ‘half\_bit’ so less confusing

assuming the get\_min\_far\_row\_idx and get\_max\_far\_row\_idx are relative to each half

changing inputs to get\_num\_minors\_per\_std\_col\_major (slrName, half\_bit rowMajor) and get\_num\_minors\_per\_bram\_content\_col\_major(slrName, half\_bit, rowMajor)

**Frame.py**: made more changes. In the basic incrementer, uses structure of:

# {

# <idcode>: {

# <rowMajors>: [ <numbers> ] // num minors per col major

# }

# }

Need to change that to

# {

# <idcode>: {

# <bottom\_half>:{

# <rowMajors>: [ <numbers> ] // num minors per col major

# }

# <top\_half>:{

# <rowMajors>: [ <numbers> ] // num minors per col major

# }

# }

# }

And see how that affects other parts of the code

Edited an incrementer function

**Bitstream\_state\_checker.py**: confirmed that the endianness it referenced from ultrascale user guide is same in 7 series.

**Extract\_clb\_col\_tile\_types**: Actually not changing the structure (is just storing slices/SLR)

ASSUMING device\_info["composition"]["slrs"].items()["max\_clock\_region\_row\_idx"] is ABOSLUTE (which seems implied). Changed