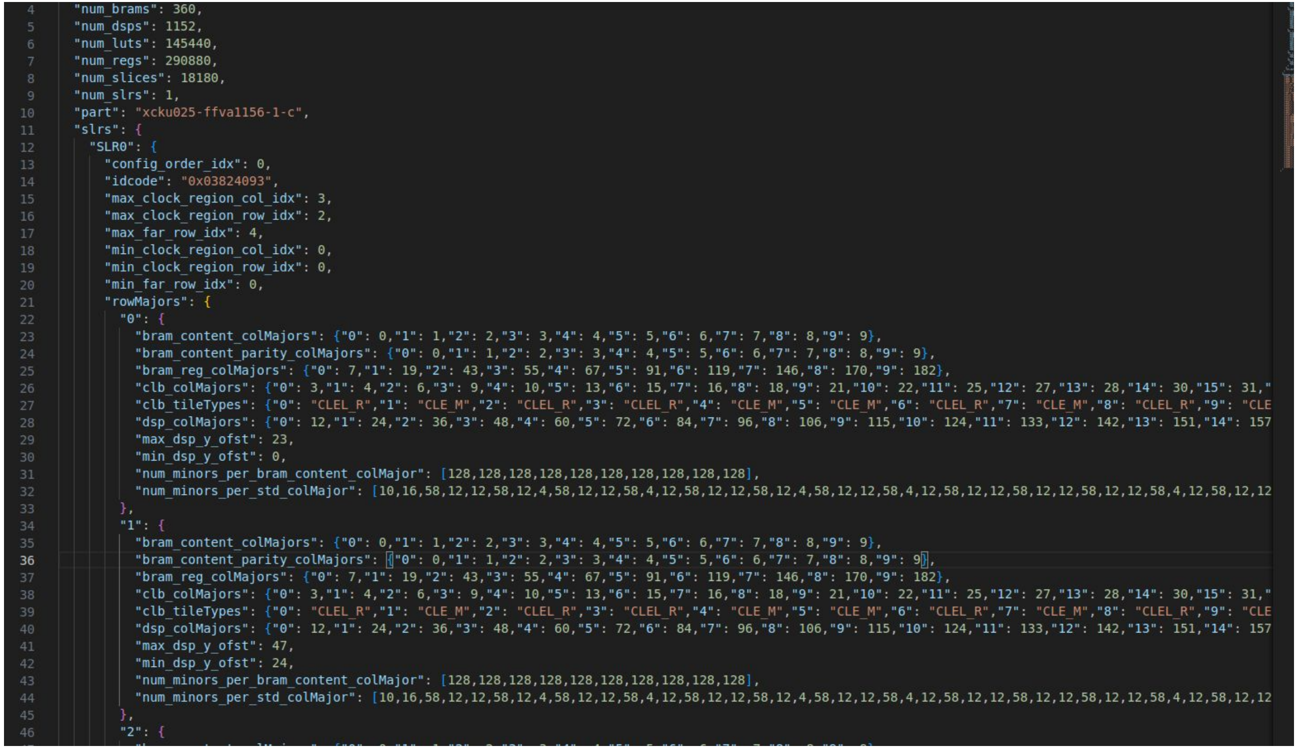
Bitfiltrator 3/21

Device Parameter Extraction (Reproduce)

✓ Used author’s scripts to extract device parameters for xcku25 and xcku35:



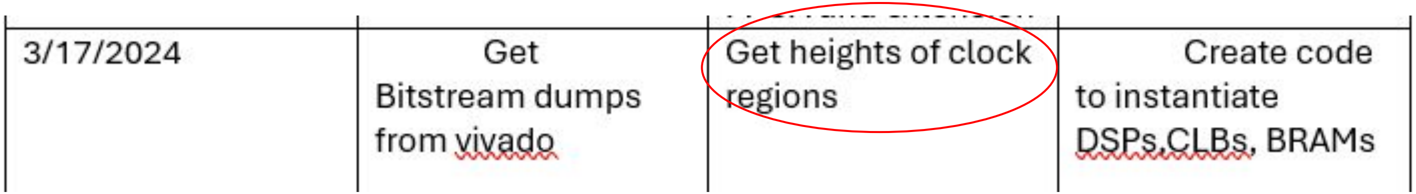
```
  "num_brams": 360,
  "num_dsps": 1152,
  "num_luts": 145440,
  "num_regs": 290880,
  "num_slices": 18180,
  "num_slrs": 1,
  "part": "xcku025-ffva1156-1-c",
  "slrs": {
    "SLR0": {
      "config_order_idx": 0,
      "idcode": "0x03824093",
      "max_clock_region_col_idx": 3,
      "max_clock_region_row_idx": 2,
      "max_far_row_idx": 4,
      "min_clock_region_col_idx": 0,
      "min_clock_region_row_idx": 0,
      "min_far_row_idx": 0,
      "rowMajors": {
        "0": {
          "bram_content_colMajors": {"0": 0,"1": 1,"2": 2,"3": 3,"4": 4,"5": 5,"6": 6,"7": 7,"8": 8,"9": 9},
          "bram_content_parity_colMajors": {"0": 0,"1": 1,"2": 2,"3": 3,"4": 4,"5": 5,"6": 6,"7": 7,"8": 8,"9": 9},
          "bram_reg_colMajors": {"0": 7,"1": 19,"2": 43,"3": 55,"4": 67,"5": 91,"6": 119,"7": 146,"8": 170,"9": 182},
          "clb_colMajors": {"0": 3,"1": 4,"2": 6,"3": 9,"4": 10,"5": 13,"6": 15,"7": 16,"8": 18,"9": 21,"10": 22,"11": 25,"12": 27,"13": 28,"14": 30,"15": 31,"
          "clb_tileTypes": {"0": "CLEL_R","1": "CLE_M","2": "CLEL_R","3": "CLEL_R","4": "CLE_M","5": "CLE_M","6": "CLEL_R","7": "CLE_M","8": "CLEL_R","9": "CLE
          "dsp_colMajors": {"0": 12,"1": 24,"2": 36,"3": 48,"4": 60,"5": 72,"6": 84,"7": 96,"8": 106,"9": 115,"10": 124,"11": 133,"12": 142,"13": 151,"14": 157
          "max_dsp_y_ofst": 23,
          "min_dsp_y_ofst": 0,
          "num_minors_per_bram_content_colMajor": [128,128,128,128,128,128,128,128,128,128],
          "num_minors_per_std_colMajor": [10,16,58,12,12,58,12,4,58,12,12,58,4,12,58,12,12,58,12,4,58,12,12,58,4,12,58,12,12,58,12,12,58,12,12,58,4,12,58,12,12
        },
        "1": {
          "bram_content_colMajors": {"0": 0,"1": 1,"2": 2,"3": 3,"4": 4,"5": 5,"6": 6,"7": 7,"8": 8,"9": 9},
          "bram_content_parity_colMajors": {"0": 0,"1": 1,"2": 2,"3": 3,"4": 4,"5": 5,"6": 6,"7": 7,"8": 8,"9": 9},
          "bram_reg_colMajors": {"0": 7,"1": 19,"2": 43,"3": 55,"4": 67,"5": 91,"6": 119,"7": 146,"8": 170,"9": 182},
          "clb_colMajors": {"0": 3,"1": 4,"2": 6,"3": 9,"4": 10,"5": 13,"6": 15,"7": 16,"8": 18,"9": 21,"10": 22,"11": 25,"12": 27,"13": 28,"14": 30,"15": 31,"
          "clb_tileTypes": {"0": "CLEL_R","1": "CLE_M","2": "CLEL_R","3": "CLEL_R","4": "CLE_M","5": "CLE_M","6": "CLEL_R","7": "CLE_M","8": "CLEL_R","9": "CLE
          "dsp_colMajors": {"0": 12,"1": 24,"2": 36,"3": 48,"4": 60,"5": 72,"6": 84,"7": 96,"8": 106,"9": 115,"10": 124,"11": 133,"12": 142,"13": 151,"14": 157
          "max_dsp_y_ofst": 47,
          "min_dsp_y_ofst": 24,
          "num_minors_per_bram_content_colMajor": [128,128,128,128,128,128,128,128,128,128],
          "num_minors_per_std_colMajor": [10,16,58,12,12,58,12,4,58,12,12,58,4,12,58,12,12,58,12,4,58,12,12,58,4,12,58,12,12,58,12,12,58,12,12,58,4,12,58,12,12
        },
        "2": {
```

✕ Somewhat lost in the steps taken…

My understanding of the steps taken

1. Vivado -> general device info (# SLRs, clock region info..?)
2. Empty bitstream and checkpoint file(?)
3. Another(?) bitstream from the DCP
4. Repeat w/ per-frame CRC
5. Parsing bitstream as per paper

Looked into Extending clock regions



3/17/2024	Get Bitstream dumps from vivado	Get heights of clock regions	Create code to instantiate DSPs,CLBs, BRAMs

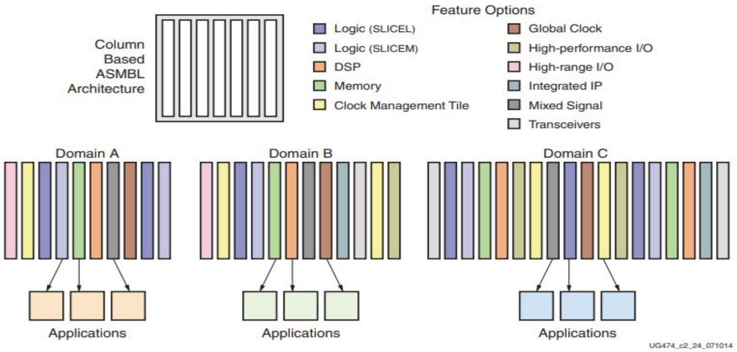
Found height for UltraScale/UltraScale+ (in paper)

Thinking about extending to Spartan 7 - is small. xc7s6?

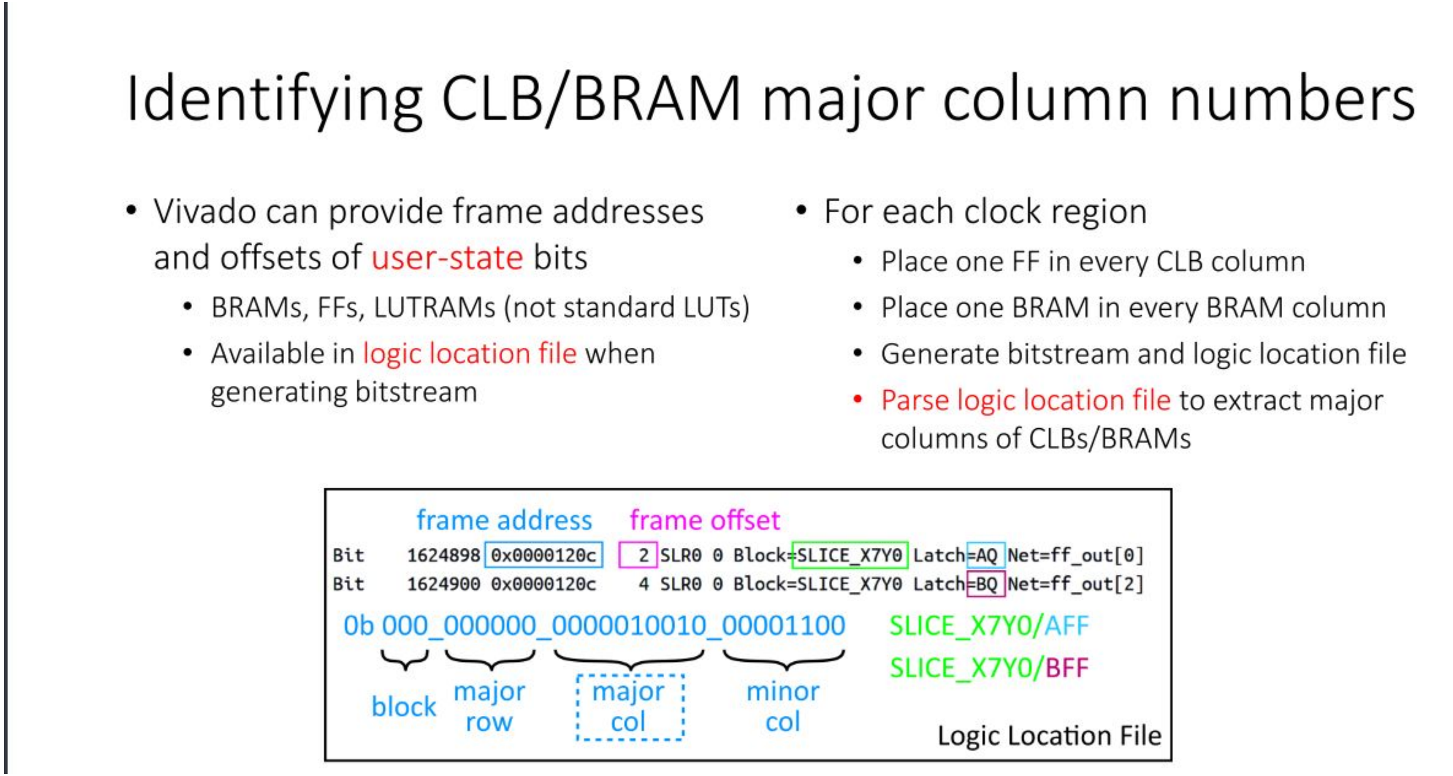
For 7 series - doesn't have clock regions. Uses ASMBL

Confused about architecture

Status: Stuck



DSP/CLB/BRAM



Identifying CLB/BRAM major column numbers

- Vivado can provide frame addresses and offsets of user-state bits
  - BRAMs, FFs, LUTRAMs (not standard LUTs)
  - Available in logic location file when generating bitstream
- For each clock region
  - Place one FF in every CLB column
  - Place one BRAM in every BRAM column
  - Generate bitstream and logic location file
  - Parse logic location file to extract major columns of CLBs/BRAMs

frame address frame offset

```
Bit     1624898 0x0000120c   2 SLR0 0 Block=SLICE_X7Y0 Latch=AQ Net=ff_out[0]
Bit     1624900 0x0000120c   4 SLR0 0 Block=SLICE_X7Y0 Latch=BQ Net=ff_out[2]
```

0b 000_000000_0000010010_00001100

block | major row | major col | minor col

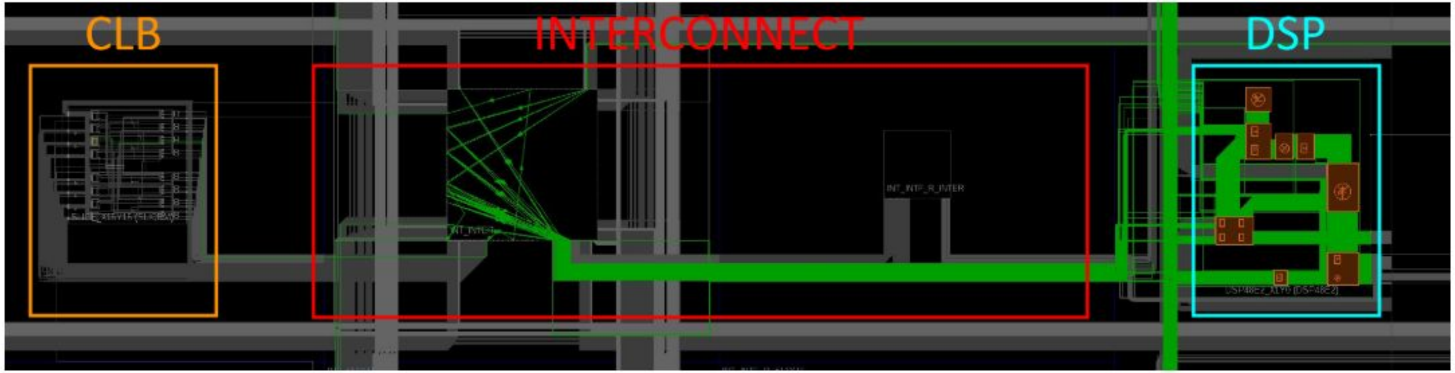
SLICE_X7Y0/AFF

SLICE_X7Y0/BFF

Logic Location File

DSP/CLB/BRAM

Identifying DSP major column numbers



- DSP inputs cannot be floating
  - Driven by constants (from CLBs)
  - Routed by interconnect
- Compare bitstream against empty one
  - Non-DSP frames also differ!

- Filter out CLB columns (now known)
- Filter out interconnect columns
  - Intuition: Interconnect columns need more minors to configure than DSP columns

→ Easy to locate and filter out