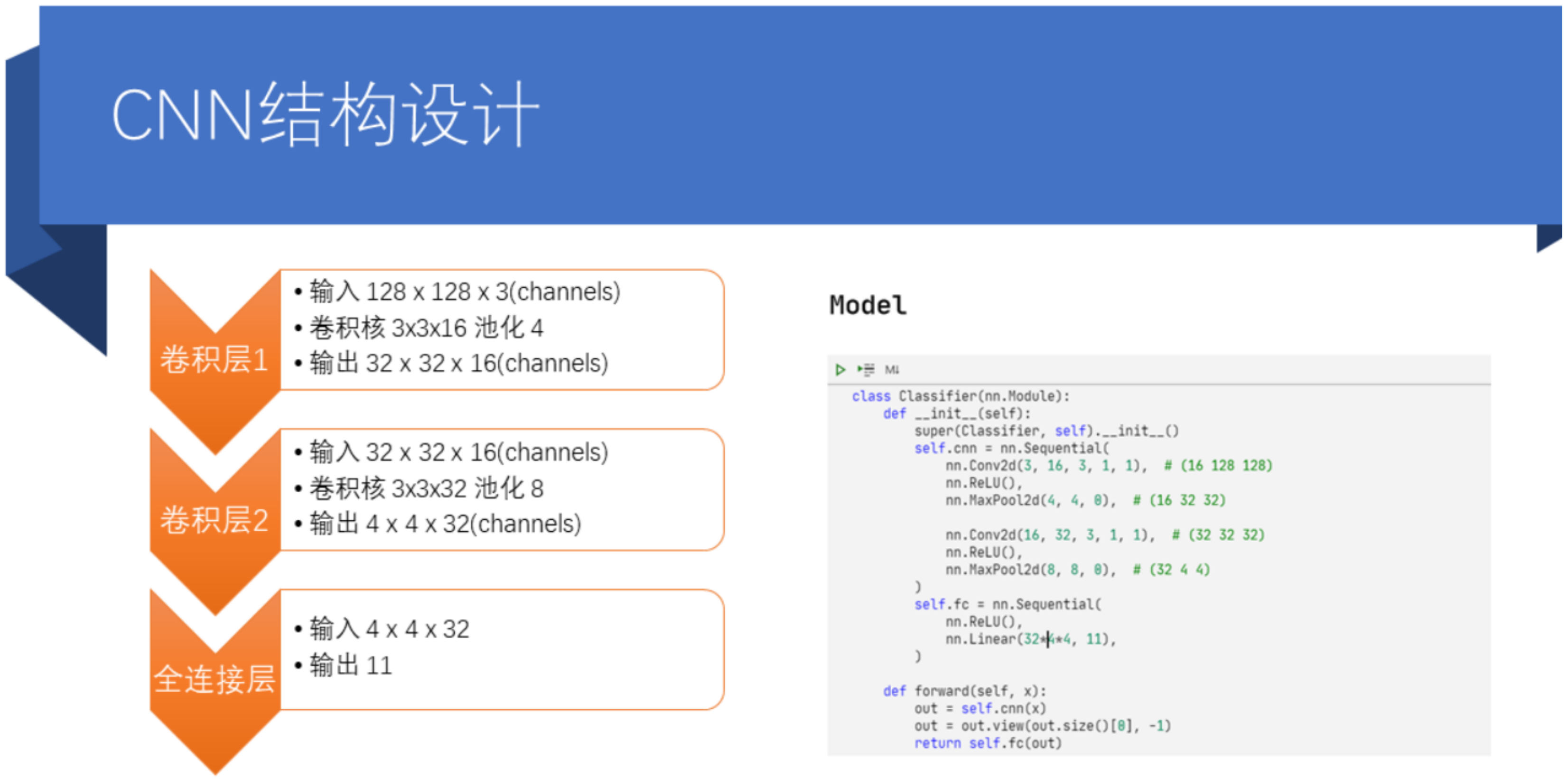
本周学习总结

许典

实验部分-实验1

- 上周的CNN模型， 在FPGA上完成了全部的结构设计和综合



CNN结构设计

卷积层1
- 输入 128 x 128 x 3(channels)
- 卷积核 3x3x16 池化 4
- 输出 32 x 32 x 16(channels)

卷积层2
- 输入 32 x 32 x 16(channels)
- 卷积核 3x3x32 池化 8
- 输出 4 x 4 x 32(channels)

全连接层
- 输入 4 x 4 x 32
- 输出 11

Model

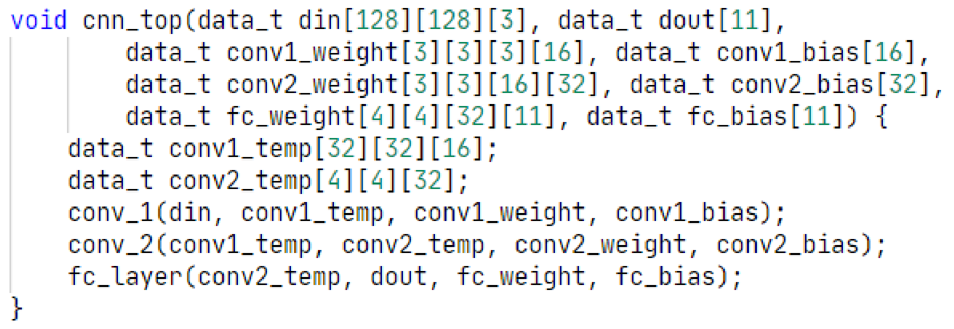
```python
class Classifier(nn.Module):
    def __init__(self):
        super(Classifier, self).__init__()
        self.cnn = nn.Sequential(
            nn.Conv2d(3, 16, 3, 1, 1),  # (16 128 128)
            nn.ReLU(),
            nn.MaxPool2d(4, 4, 0),  # (16 32 32)

            nn.Conv2d(16, 32, 3, 1, 1),  # (32 32 32)
            nn.ReLU(),
            nn.MaxPool2d(8, 8, 0),  # (32 4 4)
        )
        self.fc = nn.Sequential(
            nn.ReLU(),
            nn.Linear(32*4*4, 11),
        )

    def forward(self, x):
        out = self.cnn(x)
        out = out.view(out.size()[0], -1)
        return self.fc(out)
```

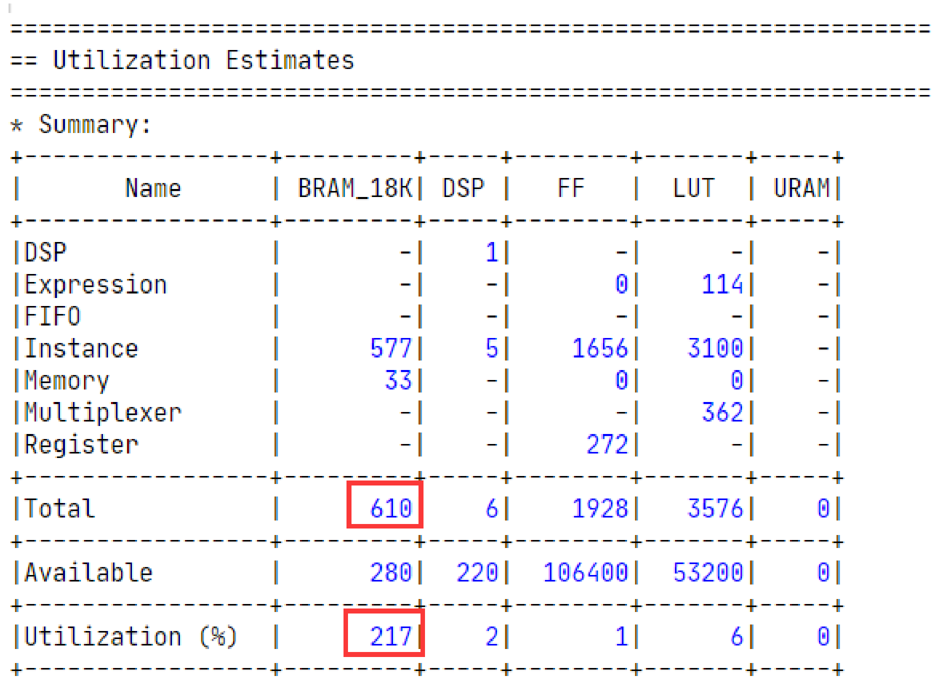
实验1-代码

顶层结构



```
void cnn_top(data_t din[128][128][3], data_t dout[11],
        data_t conv1_weight[3][3][3][16], data_t conv1_bias[16],
        data_t conv2_weight[3][3][16][32], data_t conv2_bias[32],
        data_t fc_weight[4][4][32][11], data_t fc_bias[11]) {
    data_t conv1_temp[32][32][16];
    data_t conv2_temp[4][4][32];
    conv_1(din, conv1_temp, conv1_weight, conv1_bias);
    conv_2(conv1_temp, conv2_temp, conv2_weight, conv2_bias);
    fc_layer(conv2_temp, dout, fc_weight, fc_bias);
}
```

data_t 使用float综合出来的资源占用情况



```
================================================================
== Utilization Estimates
================================================================
* Summary:
+-----------------+---------+-----+--------+-------+-----+
|       Name      | BRAM_18K| DSP |   FF   |  LUT  | URAM|
+-----------------+---------+-----+--------+-------+-----+
|DSP              |        -|    1|       -|      -|    -|
|Expression       |        -|    -|       0|    114|    -|
|FIFO             |        -|    -|       -|      -|    -|
|Instance         |      577|    5|    1656|   3100|    -|
|Memory           |       33|    -|       0|      0|    -|
|Multiplexer      |        -|    -|       -|    362|    -|
|Register         |        -|    -|     272|      -|    -|
+-----------------+---------+-----+--------+-------+-----+
|Total            |      610|    6|    1928|   3576|    0|
+-----------------+---------+-----+--------+-------+-----+
|Available        |      280|  220|  106400|  53200|    0|
+-----------------+---------+-----+--------+-------+-----+
|Utilization (%)  |      217|    2|       1|      6|    0|
+-----------------+---------+-----+--------+-------+-----+
```

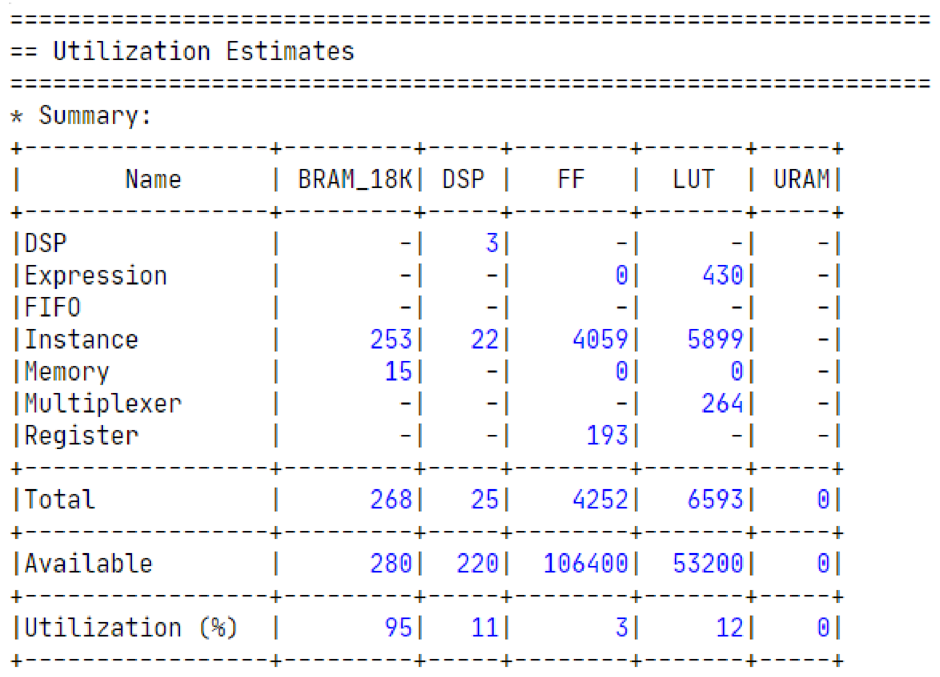
实验1-调整数据类型后

修改后数据类型

```
typedef ap_fixed<15, 4, AP_RND, AP_SAT> data_t;
```

任意精度定点数类型
数据位宽：15
整数部分位数：4
量化模式：AP_RND （指示该值应舍入到最接近ap_ [u]定点类型的可表示值）
溢出模式：AP_SAT（取到剩余位数能表示的最大的值）

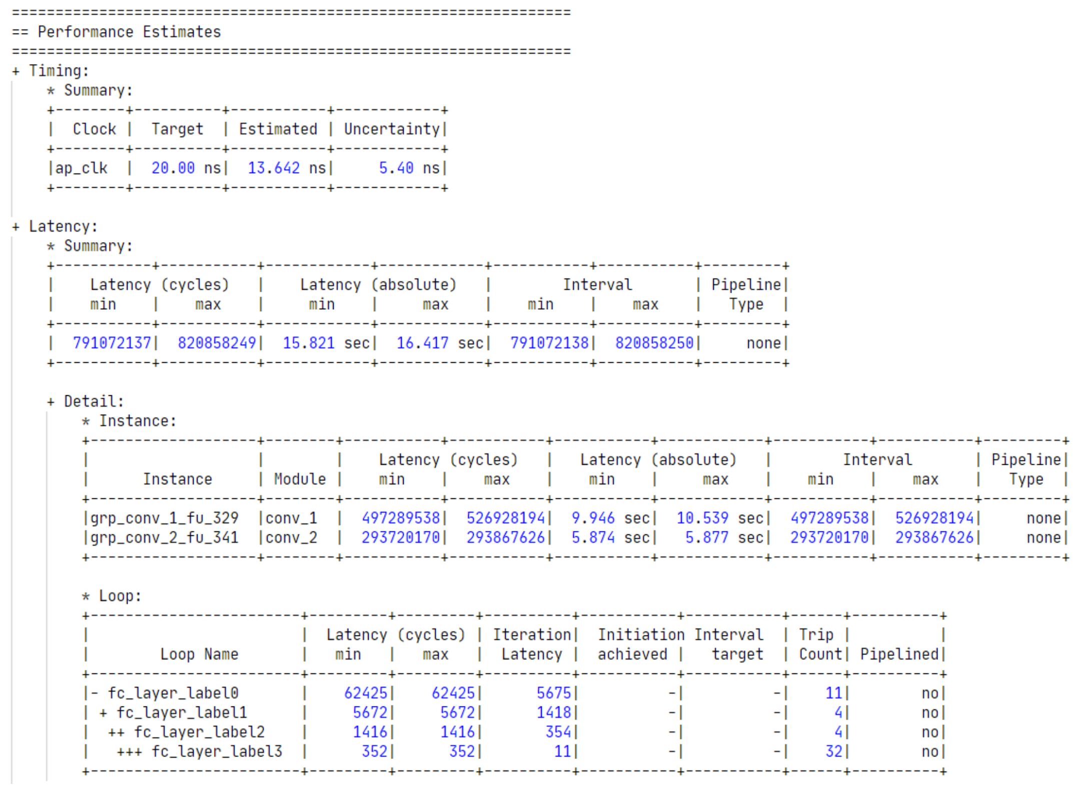
资源使用情况



```
================================================================
== Utilization Estimates
================================================================
* Summary:
+-----------------+---------+-----+--------+-------+-----+
|       Name      | BRAM_18K| DSP |   FF   |  LUT  | URAM|
+-----------------+---------+-----+--------+-------+-----+
|DSP              |        -|    3|       -|      -|    -|
|Expression       |        -|    -|       0|    430|    -|
|FIFO             |        -|    -|       -|      -|    -|
|Instance         |      253|   22|    4059|   5899|    -|
|Memory           |       15|    -|       0|      0|    -|
|Multiplexer      |        -|    -|       -|    264|    -|
|Register         |        -|    -|     193|      -|    -|
+-----------------+---------+-----+--------+-------+-----+
|Total            |      268|   25|    4252|   6593|    0|
+-----------------+---------+-----+--------+-------+-----+
|Available        |      280|  220|  106400|  53200|    0|
+-----------------+---------+-----+--------+-------+-----+
|Utilization (%)  |       95|   11|       3|     12|    0|
+-----------------+---------+-----+--------+-------+-----+
```

实验1-CNN综合结果

性能情况



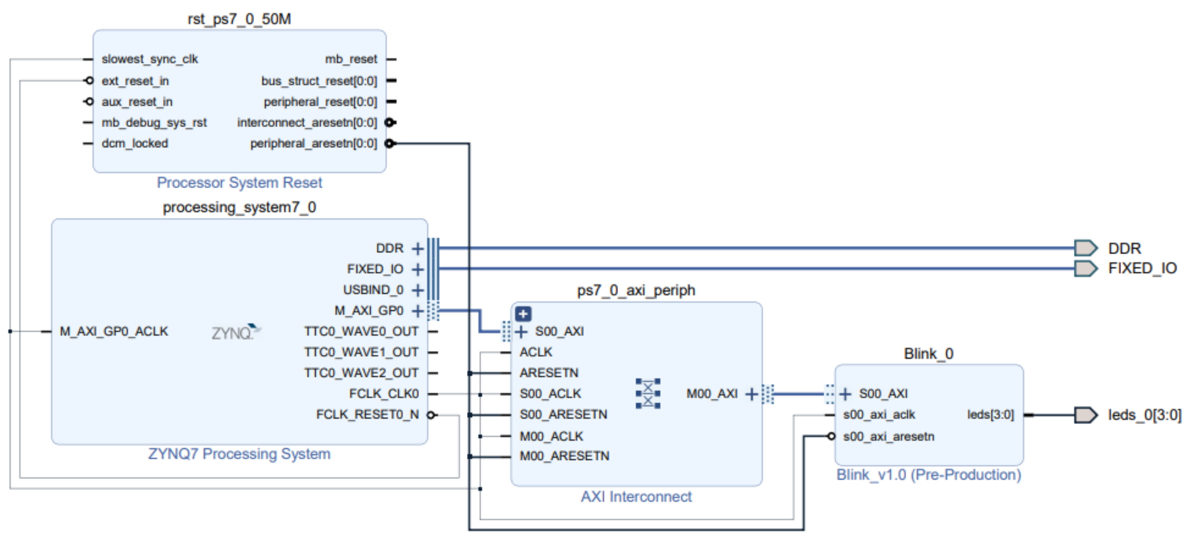
```
================================================================
== Performance Estimates
================================================================
+ Timing:
    * Summary:
    +--------+----------+-----------+------------+
    |  Clock |  Target  | Estimated | Uncertainty|
    +--------+----------+-----------+------------+
    |ap_clk  |  20.00 ns|  13.642 ns|     5.40 ns|
    +--------+----------+-----------+------------+

+ Latency:
    * Summary:
    +-----------+-----------+-----------+-----------+-----------+-----------+---------+
    |    Latency (cycles)   |   Latency (absolute)  |        Interval       | Pipeline|
    |    min    |    max    |    min    |    max    |    min    |    max    |   Type  |
    +-----------+-----------+-----------+-----------+-----------+-----------+---------+
    |  791072137|  820858249|  15.821 sec|  16.417 sec|  791072138|  820858250|     none|
    +-----------+-----------+-----------+-----------+-----------+-----------+---------+

    + Detail:
        * Instance:
        +--------------------+--------+-----------+-----------+-----------+-----------+-----------+-----------+---------+
        |                    |        |   Latency (cycles)    |  Latency (absolute)   |        Interval       | Pipeline|
        |      Instance      | Module |    min    |    max    |    min    |    max    |    min    |    max    |   Type  |
        +--------------------+--------+-----------+-----------+-----------+-----------+-----------+-----------+---------+
        |grp_conv_1_fu_329   |conv_1  |  497289538|  526928194|  9.946 sec| 10.539 sec|  497289538|  526928194|     none|
        |grp_conv_2_fu_341   |conv_2  |  293720170|  293867626|  5.874 sec|  5.877 sec|  293720170|  293867626|     none|
        +--------------------+--------+-----------+-----------+-----------+-----------+-----------+-----------+---------+

        * Loop:
        +-------------------------+---------+---------+----------+-----------+-----------+------+----------+
        |                         | Latency (cycles)  | Iteration|  Initiation Interval  | Trip |          |
        |        Loop Name        |   min   |   max   |  Latency |  achieved |   target  | Count| Pipelined|
        +-------------------------+---------+---------+----------+-----------+-----------+------+----------+
        |- fc_layer_label0        |    62425|    62425|      5675|          -|          -|    11|        no|
        | + fc_layer_label1       |     5672|     5672|      1418|          -|          -|     4|        no|
        |  ++ fc_layer_label2     |     1416|     1416|       354|          -|          -|     4|        no|
        |   +++ fc_layer_label3   |      352|      352|        11|          -|          -|    32|        no|
        +-------------------------+---------+---------+----------+-----------+-----------+------+----------+
```

实验2-验证PS和PL交互-AXILite



1. 设计了一个小的IP核（Blink），控制ZedBoard上面的Led跳动
2. 将硬件导出到Vitis中，编写程序，裸机调用
3. 使用PetaLinux将FPGA的比特流打包到BOOT.BIN，并且编写Linux驱动
4. 编写Linux 应用软件，调用系统设备

实验指导来自于ug1165

相关实验：
https://github.com/fpga-accels/dianhsu-Using_IP_with_Zynq
https://github.com/fpga-accels/dianhsu-fpga-demo1