第５回 AI エッジコンテスト
（実装コンテスト③）

TFlite delegate と rv32emc を用いた実装

課題

- 車両前方カメラの撮影動画から物体の写る矩形領域を検出し、追跡するアルゴリズムを作成する
- RISC-V をターゲットのプラットフォームに実装し、RISC-V コアを物体追跡の処理の中で使用する

山下　伸逸
ハードウェアエンジニア
主にアナログ回路設計

概要

- TensorFlow Lite (TFlite) の delegate 機構を用い、FPGA にアクセラレータを実装した
- RISC-V は rv32emc に対応する CPU core を scratch から開発した
- アクセラレータの実行制御と、物体検出結果からのトラッキング処理を1つの RISC-V で行った
- 実装は、アクセラレータ、RISC-V core とも SystemVerilog を用いた RTL 記述で行った
- 推論ネットワークは TensorFlow ssd_mobilenet_v2_320x320 を用いた
- アプリケーションは TFlite の python インターフェースを用い、RISC-V でのトラッキング処理は C 言語で開発した
- 開発したアルゴリズムは、リーダーボード上の評価 0.11 、Ultra96-V2 上での実行時間は 261 ms/frame となった

（この評価値は課題提出後にトラッキングの改良など行ったもの、参考値）

TensorFlow Lite は Google の提供する Mobile / Edge Device 用の軽量な推論プラットフォーム
TensorFlow や Keras のネットワークから軽量な FlatBuffer 形式の graph に変換する converter とその graph で推論を実行する、各種 Mobile デバイスに対応した Interpreter が提供されている
delegate 機構によって演算を外部アクセラレータに委譲することができる

推論ネットワーク

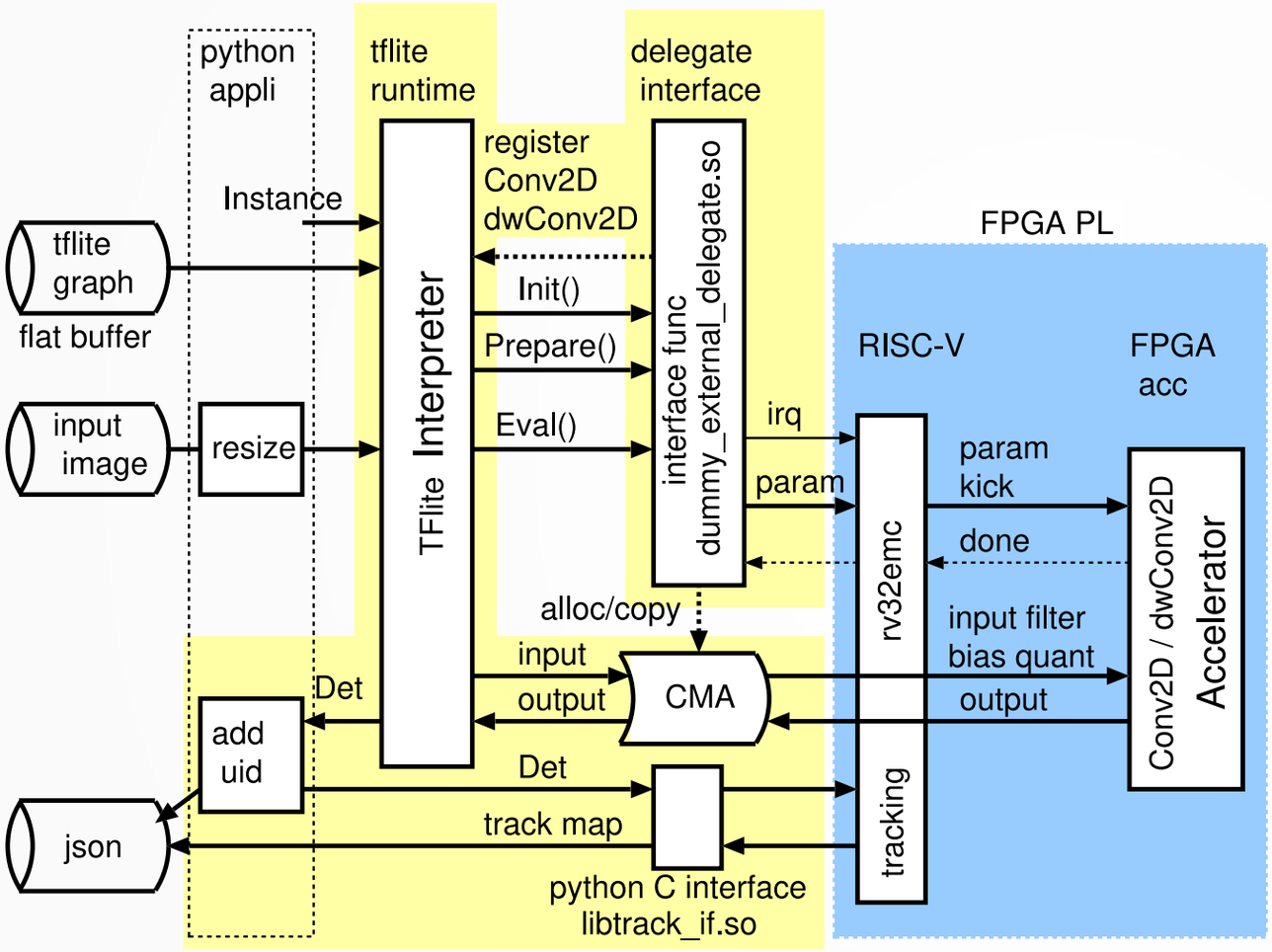
- **ネットワーク**
  - TensorFlow ssd_mobilenet_v2_320x320_coco17_tpu-8
- **90 class → 10 class で転移学習**
  - Google Colab で 250k step 学習（精度は良くなかった）
- **転移学習後に 8bit 量子化**
  - TFlite FlatBuffer 形式の graph に変換

Ultra96-V2 での TFlite benchmark (aarch64 cpu) 実行結果

```
Number of nodes executed: 102
==================== Summary by node type ========================
             [Node type]    [count]   [avg ms]    [avg %]    [cdf %]
                CONV_2D         55    323.961    84.653%    84.653%
      DEPTHWISE_CONV_2D         17     45.074    11.778%    96.431%
TFLite_Detection_PostProcess     1      9.418     2.461%    98.892%
                     ADD         10     3.018     0.789%    99.681%
                QUANTIZE         1      0.870     0.227%    99.908%
                LOGISTIC         1      0.149     0.039%    99.947%
              DEQUANTIZE         2      0.113     0.030%    99.977%
           CONCATENATION         2      0.054     0.014%    99.991%
                 RESHAPE         13     0.035     0.009%   100.000%
                               total  382.741 ms
```

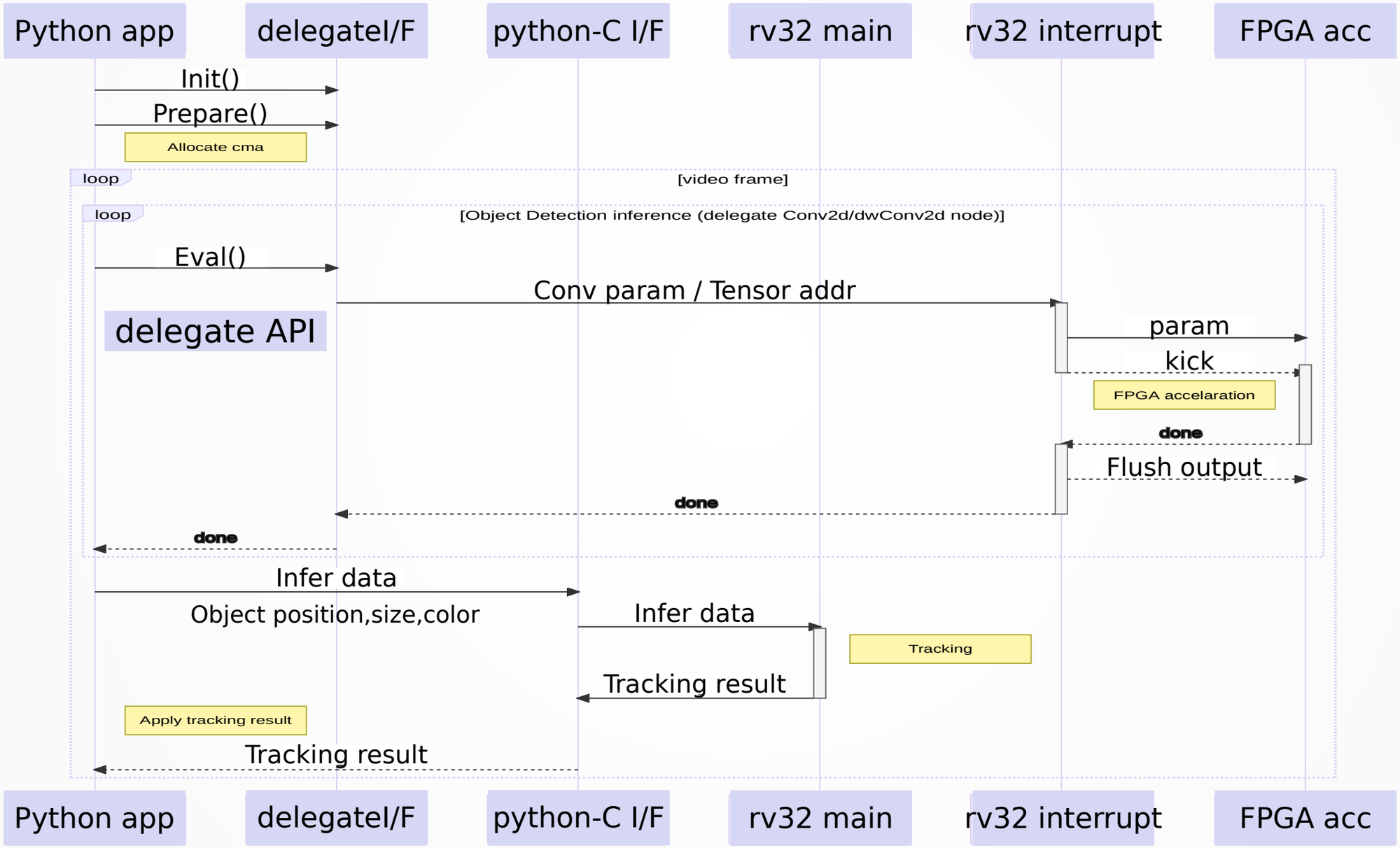
Conv2D,dwConv2D で96 .4%を占める
→ この2つの演算を FPGA に delegate する

システム構成



- TFlite Interpreter の delegate API を用いて FPGA PL に演算を渡す
- delegate する演算種別 (Conv2D, depthwiseConv2D) を登録する
- Interpreter は graph を実行し、登録した演算のみインターフェース関数に渡して実行を委譲
- インターフェース関数には Conv 演算パラメータと、Tensor へのポインターが渡され、これを RISC-V に渡して ハードウェアを kick し、演算終了を待つ
- Interpreter が推論ネットワークの Object Detection 結果を出した後、これを python の C-interface を介して FPGA PL に送り、RISC-V がトラッキング処理を行う。
- トラッキング処理結果を再び python に返し、json に反映する
- FPGA からは Linux の VM 領域がアクセスしにくいので、CMA 領域を介してやり取りする

RISC-V と推論実行アプリのシーケンス



rv32emc の開発

- 組み込み用途向けの EMC (32bit 16 Register, Mul/Div, Compressed 命令) の構成で CPU core を開発した
- Fetch/Decode/Exec/MemoryAccess/Writeback の5段パイプライン構成とした

開発の手順

1 CPU のハード構成を設計し、パイプラインまでエミュレートした ISS をC言語で作成

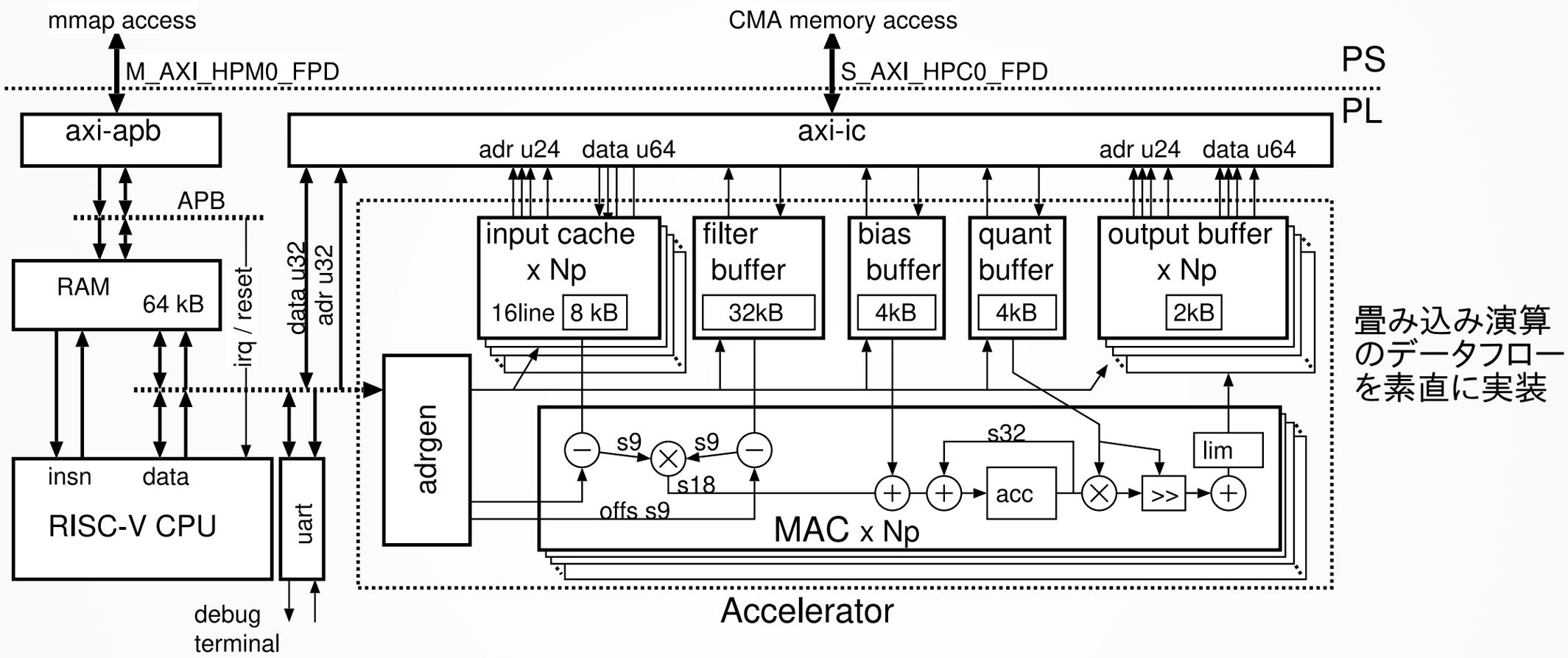
2 risc-v のクロス gdb でのテストプログラム実行結果をリファレンスとして ISS をデバッグ

3 作成した ISS をリファレンスとして HDL(SystemVerilog) を記述、ISS の実行トレースと HDL の論理シミュレーション (xsim) 結果をつき合わせて HDL をデバッグ

   ISS 用に作成した命令のテーブルを使用して HDL の命令テーブルを自動生成するようにしてミスを防いだ

4 割り込み機構を実装、タイマー (mtime) でタイマー割り込みができるようにして、Vivado で論理合成し FPGA にロード、実機でターミナルをつないで動作確認

ハードウエア構成

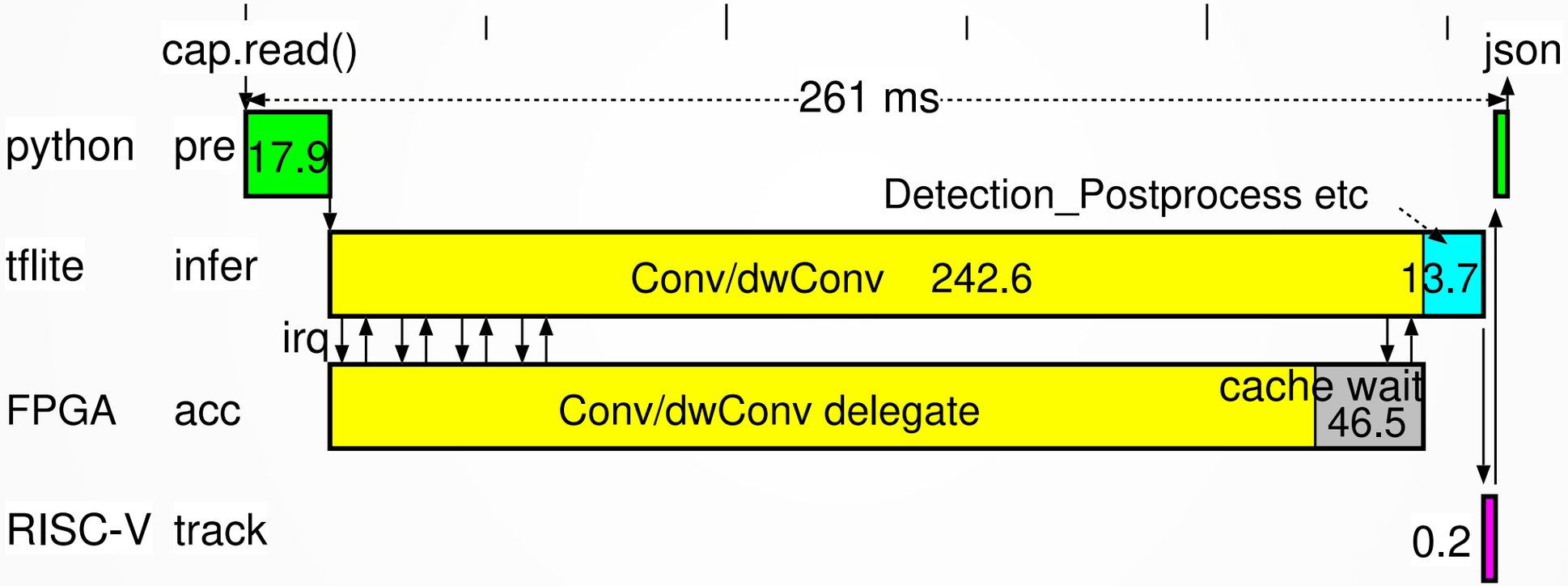


- Conv 演算（MAC）は、Np 個並列に実装し、input, output のデータアクセスには Np 個並列にキャッシュメモリを設けた
- filter, bias, quant は Conv 演算で共通、それぞれ 1 個のバッファメモリを設けた
- PS から AXI-APB bridge を介して RISC-V の RAM をアクセスする。実行バイナリを RAM にロードし、reset を解除することで CPU を起動する
- APB から割り込みを発生し、アクセラレータを kick する。アクセラレータ制御を割り込み処理で行い、トラッキング処理を並行処理する

実行結果

MAC 並列数　Np = 42
演算クロック周波数 = 125 MHz

推論処理時間内訳



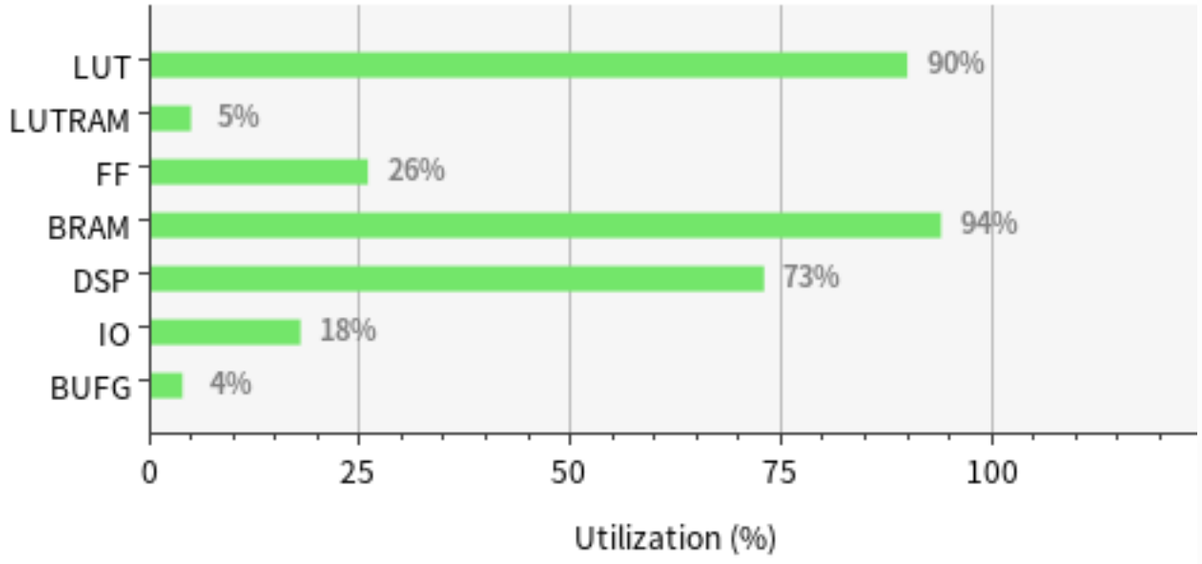
推論実行に約 261ms/frame を要した
RISC-V によるトラッキング処理は約 0.2 ms と小さい

まとめ

- TFlite の delegate 機構を用いた FPGA アクセラレータを開発し、動作を確認することができた。
- RISC-V (rv32emc) を RTL で実装し、FPGA に組み込んで、アクセラレータ制御とトラッキング処理に用いて動作を確認した。
- リーダーボードの評価値は 0.11 と低いが、mobilenet_v2 の転移学習結果の検出精度が低い様で、トラッキングアルゴリズムも簡易なためトラッキングが継続できない様子が見られる。
- 速度は 261 ms/frame と遅い。MAC の並列数が 42 にとどまることが大きい。今回の並列化は1次元であるが、output チャンネルも 4 ～ 8 並列として2次元の並列化で FPGA の DSP リソースの活用を行うのが効果が大きいと思われる。
- 今回初期化処理 Prepare() で filter Tensor を CMA にコピーしているので、このときにチャンネル並列に合わせた filter 順序に並べ替えることで、最小限のハード追加で2次元化できそうである。

以上

Resource	Utilization	Available	Utilization %
LUT	63426	70560	89.89
LUTRAM	1416	28800	4.92
FF	36744	141120	26.04
BRAM	204	216	94.44
DSP	263	360	73.06
IO	15	82	18.29
BUFG	7	196	3.57



Utilization

Np : 42
clock: 125 MHz

Block Design に２つの RTL ブロックを読み込んで接続

Accelerator

RISC-V