RISC-V®

Graphics and ML SIG Meeting
May 26, 2022
10:05am PDT

https://github.com/riscv-admin/graphics

@risc_v

Only RISC-V Members May Attend

- Non-members are asked to please leave except for Joint Working Groups (JWG).
- Members share IP protection by virtue of their common membership agreement. Non-members being present jeopardizes that protection. Joint working groups (JWG) agree that any IP discussed or worked on is fully open source and unencumbered as per the policy.
- It is easy to become a member. Check out riscv.org/membership
- If you need work done between non-members or or other orgs and RISC-V, please use a joint working group (JWG).
  - used to allow non-members in SIGs but the SIGs purpose has changed.
- Please put your name and company (in parens after your name) as your zoom name. If you are an individual member just use the word "individual" instead of company name.
- Non-member guests may present to the group but should only stay for the presentation. Guests should leave for any follow on discussions.

Antitrust Policy Notice

RISC-V International meetings involve participation by industry competitors, and it is the intention of RISC-V International to conduct all its activities in accordance with applicable antitrust and competition laws. It is therefore extremely important that attendees adhere to meeting agendas, and be aware of, and not participate in, any activities that are prohibited under applicable US state, federal or foreign antitrust and competition laws.

Examples of types of actions that are prohibited at RISC-V International meetings and in connection with RISC-V International activities are described in the RISC-V International Regulations Article 7 available here: https://riscv.org/regulations/

If you have questions about these matters, please contact your company counsel.

Collaborative & Welcoming Community

RISC-V is a free and open ISA enabling a new era of processor innovation through open standard collaboration. Born in academia and research, RISC-V ISA delivers a new level of free, extensible software and hardware freedom on architecture, paving the way for the next 50 years of computing design and innovation.

We are a transparent, collaborative community where all are welcomed, and all members are encouraged to participate. We are a continuous improvement organization. If you see something that can be improved, please tell us. help@riscv.org

We as members, contributors, and leaders pledge to make participation in our community a harassment-free experience for everyone.

https://riscv.org/community/community-code-of-conduct/

Conventions

- **For one hour meetings, please start at 5 after the start time** in order to allow people going to other meetings have time for a short break between meetings. 30 minute meetings start on time.
- Unless it is a scheduled agenda topic, we don’t solve problems or detailed topics in most meetings unless specified in the agenda because we don’t often have enough time to do so and it is more efficient to do so offline and/or in email. We identify items and send folks off to do the work and come back with solutions or proposals.
- If some policy, org, extension, etc. can be doing things in a better way, help us make it better. Do not change or not abide by the item unilaterally. Instead let's work together to make it better.
- Please conduct meetings that accommodates the virtual and broad geographical nature of our teams. This includes meeting times, repeating questions before you answer, at appropriate times polling attendees, guide people to interact in a way that has attendees taking turns speaking, …
- Where appropriate and possible, meeting minutes will be added as speaker notes within the slides for the Agenda

Agenda

- Our zoom link has already changed
- Discuss whether to focus our effort on Winograd or direct convolution
- Supporting for the F(6x6, 3x3) Winograd formulation
- ML: call for dashboard updates https://github.com/riscv-admin/graphics/blob/main/ml-dashboard.adoc

Winograd

Resources:

https://openaccess.thecvf.com/content_cvpr_2016/papers/Lavin_Fast_Algorithms_for_CVPR_2016_paper.pdf
Lavin and Gray’s paper, rescuing the concept from Shmuel Winograd

https://github.com/andravin/wincnn
Supplementary material, including a python tool for generating matrices for arbitrary number of outputs and kernel sizes.

https://openreview.net/pdf?id=H1ZaRZVKg
A discussion on numerical stability for big tile sizes.

https://arxiv.org/abs/1512.03385
Resnet paper, with Resnet18 convolutions listed in Table 1.

Winograd recipe

Matrices given by formulation

$$Y = A^T \left[\underbrace{[GgG^T]}_{\text{Kernel}} \odot \underbrace{[B^T dB]}_{\text{Input tile}} \right] A \qquad (8)$$

Winograd formulation

F(m x m, r x r)

m x m: is the output tile size

r x r: kernel size

Input tile size is implicitly (m + r - 1) x (m + r - 1)

F(6x6, 3x3)

6 x 6: is the output tile size
3 x 3: kernel size
Input tile size is implicitly 8x8

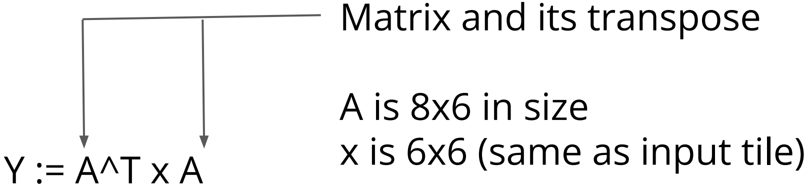
As contributed by Andrew Lavin to NNPACK
https://github.com/Maratyszcza/NNPACK/issues/8

```
AT =
⎡1   1   1    1     1    32   32     0⎤
⎢                                     ⎥
⎢0   1   -1   2    -2    16   -16    0⎥
⎢                                     ⎥
⎢0   1   1    4     4    8      8    0⎥
⎢                                     ⎥
⎢0   1   -1   8    -8    4     -4    0⎥
⎢                                     ⎥
⎢0   1   1    16   16    2      2    0⎥
⎢                                     ⎥
⎣0   1   -1   32   -32   1     -1    1⎦
```

```
G =
⎡ 1       0       0    ⎤
⎢                      ⎥
⎢-2/9   -2/9    -2/9   ⎥
⎢                      ⎥
⎢-2/9    2/9    -2/9   ⎥
⎢                      ⎥
⎢1/90   1/45    2/45   ⎥
⎢                      ⎥
⎢1/90   -1/45   2/45   ⎥
⎢                      ⎥
⎢1/45   1/90    1/180  ⎥
⎢                      ⎥
⎢1/45   -1/90   1/180  ⎥
⎢                      ⎥
⎣ 0       0       1    ⎦
```

```
BT =
⎡1    0    -21/4     0     21/4     0     -1   0⎤
⎢                                               ⎥
⎢0    1      1    -17/4   -17/4     1      1   0⎥
⎢                                               ⎥
⎢0   -1      1    17/4    -17/4    -1      1   0⎥
⎢                                               ⎥
⎢0   1/2    1/4   -5/2    -5/4      2      1   0⎥
⎢                                               ⎥
⎢0  -1/2    1/4    5/2    -5/4     -2      1   0⎥
⎢                                               ⎥
⎢0    2      4    -5/2     -5      1/2     1   0⎥
⎢                                               ⎥
⎢0   -2      4     5/2     -5     -1/2     1   0⎥
⎢                                               ⎥
⎣0   -1      0    21/4      0    -21/4   0     1⎦
```

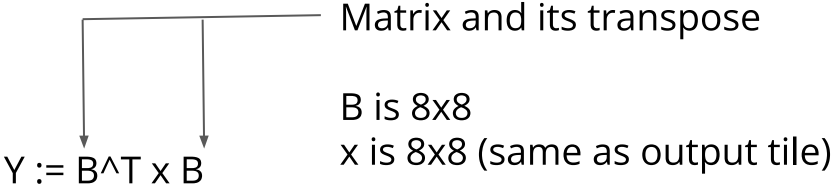
What are Winograd specific operations?



Matrix and its transpose

A is 8x6 in size
x is 6x6 (same as input tile)

Y := A^T x A



Matrix and its transpose

B is 8x8
x is 8x8 (same as output tile)

Y := B^T x B

Winograd or direct convolution?

- All methods needed! Unfortunately…

- Most common cases handled optimally with F(6x6, 3x3).
  Stride is 1, kernel size is 3x3
  That handles all of the Resnet18 convolutions!

- Fallback of Winograd is a direct convolution engine, can share logic with pooling

- Fallback of direct convolution engine is vector code, for cases using dilation or 3D convolution

Open floor session

Backup Slides