**CGRA\_V3.2.5**

**OVERALL VWR2A**

RCs have 64 configuration words. Each RC has 3 input/output VWRs (A, B and C). Each column has a Load and Store Unit (LSU) that takes care of moving the data between the scratchpad memory (SP) and the Very Wide Register (VWR).

**KERNEL MEMORY**

VWR2A can have multiple kernels loaded into its instruction memory (IMEM). Each specialized slot has an IMEM of 64 words, while the global IMEM can store 512 words. To keep track of which kernel instructions start at which IMEM address location, we use a kernel memory that lists up to 16 possible kernels to execute. Each kernel is set up with a configuration word:

|  |  |  |  |
| --- | --- | --- | --- |
| **CGRA CONFIGURATION WORD FORMAT** | | | |
| SRF ADDRESS | # COLUMNS (one-hot encoding) | KERNEL START ADDRESS | RCS/CCS/MUX/LSU  # INSTR. |
| 20:17 | 16:15 | 14:6 | 5:0 |
| TOTAL: 21 bits | | | |

Parameters:

* # INSTR (6 bits): Number of instructions the kernel takes in the IMEM of each specialized slot. The maximum is 64 because that is the number of words in the local IMEMs of the specialized slots.
* KERNEL START ADDRESS (9 bits): start address of the kernel in the IMEM. Ranges from 0 to 511.
* # COLUMNS (2 bits). One-hot encoding of the columns that the kernel runs on. “01” means column 0, “10” means column 1, and “11” means both.
* SRF ADDRESS (4 bits): Address of the Scratchpad Memory (SPM) that the scalar register file (SRF) of the kernel occupies. Ranges from 0 to 15.

Example:

0000 01 000000000 101011

SPM Address 0 --- Column 0 --- IMEM address 0 ---- 43 instructions

**CGRA DMA**

Controlled through CGRA APB register

REGISTERS:

0: Core 0 kernel id request

1: Core 1 kernel id request

2: DMA address pointer

3: DMA transfer type: 1 bit write + 1 bit read + 1 bit push enable (0=disable, 1=enable), push line to SP in case it is not full) + size 15 bits (max=8192, 32kB)

4: reserved

5: reserved

6: events register: indicates interrupt from DMA or kernels

7: cgra status: 2 bits DMA pending request + core and kernel id currently runnning on the columns (0=free)

**CGRA LCU**

The Loop-Control Unit (LCU) is responsible for updating the program counter of the CGRA. It has branch instructions that branch to the immediate value (except in the case of JUMP, that branches to the sum of the MUXA and MUXB results). It also has an ALU whose result is stored in one of the registers when RF\_WE is enabled. It also issues the EXIT command at the end of every kernel to wake up the host processor and put the CGRA to sleep.

Parameters:

* Immediate: 6-bit value of an IMEM address to branch to in all branch operations except JUMP. Also can be passed into the ALU for logic operations through MUX\_A.
* RF\_WSEL (2 bits): choose one of the 4 local registers to write to
* RF\_WE (1 bit): enable writing ALU result to the chosen register
* ALU\_OP (4 bits): ALU operation to run (see possibilities below)
* BR\_MODE (1 bit): 0 to control the program counter, 1 to control RC datapath
* MUX{A/B}\_SEL (3 bits) : Select inputs to the ALU.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **CGRA LCU INSTRUCTION FORMAT** | | | | | | |
| MUXA\_SEL | MUXB\_SEL | BR\_MODE | ALU\_OP | RF\_WE | RF\_WSEL | IMMEDIATE |
| 19:17 | 16:14 | 13 | 12:9 | 8 | 7:6 | 5:0 |
| TOTAL: 20 bits | | | | | | |

BR\_MODE: CC (0) loop control (CCs alu) or RC (1) data control (RCs alu)

MUXA\_SEL:

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 |
| R0 | R1 | R2 | R3 | SRF | LAST | 0 | IMM |

MUXB\_SEL:

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 |
| R0 | R1 | R2 | R3 | SRF | LAST | 0 | 1 |

R[0-3]: local registers

IMM: immediate value extended to datapath width

LAST: VWR\_SLICE-1

COL\_ID: column ID (0 to N\_COL-1)

|  |  |  |
| --- | --- | --- |
| **CGRA LCU OPERATIONS** | | |
| 0 | NOP | no operation |
| 1 | SADD | signed addition |
| 2 | SSUB | signed subtraction |
| 3 | SLL | shift left logical |
| 4 | SRL | shift right logical |
| 5 | SRA | shift right arithmetic |
| 6 | LAND | logical AND |
| 7 | LOR | logical OR |
| 8 | LXOR | logical XOR |
| 9 | BEQ | branch if equal (a == b) |
| 10 | BNE | branch if not equal (a != b) |
| 11 | BGEPD | branch if greater or equal ((a--)>= b) with pre-decrement |
| 12 | BLT | branch if less than (a < b) |
| 13 | JUMP | jump to ina+inb |
| 14 | EXIT | kernel exit instruction |
| 15 | NOP | no operation |

**CGRA LSU**

The Load-Store Unit (LSU) is responsible for generating the bank of the SPM to write into a VWR, or to write from a VWR back to the SPM. It also takes care of shuffling the values in VWRs A and B and storing the result into VWR C. It has 8 local registers, and Register 7 holds the line of the SPM for LOAD and STORE operations. Register 7 is initialized to the SRF address.

Parameters:

* RF\_WSEL (3 bits): One of 8 LSU registers to write ALU result to
* RF\_WE (1 bit): Enable writing to LSU registers
* OP\_1 (3 bits): ALU operation to perform between MUXA and MUXB results
* MUX{A/B}\_SEL (4 bits) : select inputs to ALU
* VWR\_SEL/SHUF\_OP (3 bits): Depending on the input OP\_2, either choose a VWR/SRF to write to/from, or select a shuffle operation.
  + In the case of VWR LOAD/STORE (2 bits):
    - 0: VWR A
    - 1: VWR B
    - 2: VWR C
    - 3: SRF
  + In the case of shuffling (3 bits):
    - 0: VWRA and VWRB interleaving upper part
    - 1: VWRA and VWRB interleaving lower part
    - 2: VWRA and VWRB even indexes
    - 3: VWRA and VWRB odd indexes
    - 4: VWRA and VWRB concatenated bit reversal upper part
    - 5: VWRA and VWRB concatenated bit reversal lower part
    - 6: VWRA and VWRB concatenated slice circular shift up upper part
    - 7: VWRA and VWRB concatenated slice circular shift up lower part

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **CGRA LSU INSTRUCTION FORMAT** | | | | | | |
| OP\_2 | VWR\_SEL  SHUF\_OP | MUXA\_SEL | MUXB\_SEL | OP\_1 | RF\_WE | RF\_WSEL |
| 19:18 | 17:15 | 14:11 | 10:7 | 6:4 | 3 | 2:0 |
| TOTAL: 20 bits | | | | | | |

MUX\_A\_SEL select shuffle type

|  |  |  |
| --- | --- | --- |
| **CGRA LSU OPERATIONS 2** | | |
| 0 | NOP | no operation |
| 1 | LOAD | Load line from SP to VWR |
| 2 | STORE | Store VWR to SP line |
| 3 | SHUFFLE | Shuffle data from VWR A and B and result is stored in C |

MUXA\_SEL:

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 0-7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 |
| R0-R7 | SRF | 0 | 1 | 2 | 0 | 0 | 0 | 0 |

MUXB\_SEL:

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 0-7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 |
| R0-R7 | SRF | 0 | 1 | 2 | 0 | 0 | 0 | 0 |

|  |  |  |
| --- | --- | --- |
| **CGRA LSU OPERATIONS 1** | | |
| 0 | LAND | logical AND |
| 1 | LOR | logical OR |
| 2 | LXOR | logical XOR |
| 3 | SADD | signed addition |
| 4 | SSUB | signed subtraction |
| 5 | SLL | shift left logical |
| 6 | SRL | shift right logical |
| 7 | BITREV | bit reversal operation |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **CGRA RCS INSTRUCTION FORMAT** | | | | | | |
| MUXA\_SEL | MUXB\_SEL | OP\_MODE | ALU\_OP | MUXF\_SEL | RF\_WE | RF\_WSEL |
| 17:14 | 13:10 | 9 | 8:5 | 4:2 | 1 | 0 |
| TOTAL: 18 bits | | | | | | |

MUXA\_SEL:

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 0 | 1 | 2 | 3 | 4-5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14-15 |
| VWRA | VWRB | VWRC | SRF | R0-1 | RCT | RCB | RCL | RCR | 0 | 1 | MAX INT | MIN INT | - |

MUXB\_SEL:

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 0 | 1 | 2 | 3 | 4-5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14-15 |
| VWRA | VWRB | VWRC | SRF | R0-1 | RCT | RCB | RCL | RCR | 0 | 1 | MAX INT | MIN INT | - |

MUXF\_SEL:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 0 | 1 | 2 | 3 | 4 | 5-7 |
| OWN | RCT | RCB | RCL | RCR | - |

VWR\_A/B : data from VRW\_A/B

R[0-1]: local registers

RC[L/R/T/B] : **R**econfigurable **C**ell **L**eft/**R**ight/**T**op/**B**ottom result (from previous clock cycle)

MAX/MIN INT: max/min signed value for RCs datapath

OP\_MODE: 0:32b, 1:16b

RF\_WE:

The alu output is written to the RF\_SEL register if RF\_WE equals 1. It is not always needed to store the output back to the register file, because each RC is connected to the neighbored RCs through a register. The data can also be written to the VWRs (controlled by the CGRA MUX).

RF\_WSEL:

Each RC has 2 registers. This field is used to select to which register the output should be written to.

|  |  |  |
| --- | --- | --- |
| **CGRA RCS OPERATIONS** | | |
| 0 | NOP | no operation |
| 1 | SADD | signed addition |
| 2 | SSUB | signed subtraction |
| 3 | SMUL | signed multiplication |
| 4 | SDIV | signed division (reserved but not implemented) |
| 5 | SLL | shift left logical |
| 6 | SRL | shift right logical |
| 7 | SRA | shift right arithmetic |
| 8 | LAND | logical AND |
| 9 | LOR | logical OR (INA) |
| 10 | LXOR | logical XOR |
| 11 | INB\_SF\_INA | INA out if sign flag = 1 else INB out |
| 12 | INB\_ZF\_INA | INA out if zero flag = 1 else INB out |
| 13 | FXP\_MUL | fixed point multiplication (1b sign + half\_datapath\_width integer + half\_datapath\_width-1 decimal) |
| 14 | FXP\_DIV | fixed point division (reserved but not implemented) |
| 15 | NOP | no operation |

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **CGRA VWR\_MUX INSTRUCTION FORMAT** | | | | | | | | | |
| MUXA\_SEL | MUXB\_SEL | OPS | RF\_WE | RF\_WSEL | SCR\_WE | SCR\_WD | SCR\_SEL | VWR\_SEL | VWR\_ROW\_WE |
| 26:23 | 22:19 | 18:16 | 15 | 14:12 | 11 | 10:9 | 8:6 | 5:4 | 3:0 |
| TOTAL: 27 bits | | | | | | | | | |

RF\_WE: local register file write enable

RF\_WSEL: select local register to write to

SCR\_WE: scalar register file write enable

SCR\_WD: scalar register file write data selection between CONTROL\_CELL, RC0\_CELL, MUX\_CELL, and LSU\_CELL

VWR\_SEL: select the register to write (A, B or C)

VWR\_ROW\_WE: write enable to VWR A/B/C (one bit per slice/row)

MUXA\_SEL:

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 0-7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 |
| R0-R7 | SRF | 0 | 1 | 2 | ½ VWR\_SIZE | LAST\_VWR | 0 | 0 |

MUXB\_SEL:

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 0-7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 |
| R0-R7 | SRF | 0 | 1 | 2 | ½ VWR\_SIZE | LAST\_VWR | 0 | 0 |

R0: this register holds the value to select the vwr entry passed to the RCs datapath

R5: VWR\_A mask (vwr\_sel=R0&R5)

R6: VWR\_B mask (vwr\_sel=R0&R6)

R7: VWR\_C mask (vwr\_sel=R0&R7)

SRF: scalar register file

|  |  |  |
| --- | --- | --- |
| **CGRA VWR\_MUX OPERATIONS** | | |
| 0 | NOP | no operation |
| 1 | SADD | signed addition |
| 2 | SSUB | signed subtraction |
| 3 | SLL | shift left logical |
| 4 | SRL | shift right logical |
| 5 | LAND | logical AND |
| 6 | LOR | logical OR |
| 7 | LXOR | logical XOR |