Miaow - Unit Specification

February 25, 2013

1 Introduction

Between stage flops are owned by the unit for whose signals are inputs.

2 Units

2.1 Fetch

2.1.1 Wavefront identifiers - WF_ID and WF_TAG

WF_ID: Identifies a wavefront within the Compute Unit, used to address values that are private for each wavefront.

WF_TAG: Global identifier for a wavefront on the kernel

2.1.2 INTERFACE

WF_DISPATCH: Dispatch enable

WF_TAG_DISPATCH: Global identifier for a wavefront on the kernel

WF_SIZE_DISPATCH: Number of threads in wavefront

WF_PC: Start PC of wavefront

VGPR_BASE: VGPR base address for the wavefront

SGPR_BASE: SGPR base address for the wavefront

LDS_BASE: LDS base address for the wavefront

WFID_DONE: WF_ID of wavefront halted

HALT: End of wavefront received from Issue stage

ACK: Ack from instruction memory

WF_ID: Global identifier for a wavefront on the kernel

BUFF_RD: Read enable for instruction memory

BUFF_ADDR: PC of next instruction to be read

MASK_INIT: Start PC of wavefront

BASE_WR: Write enable for base values and mask

BUFF_TAG:

BUFF_TAG[39]	Set if its first instruction of the wavefront
BUFF_TAG[38:7]	PC of instruction sent to memory
BUFF_TAG[6:0]	WF_ID of instruction sent to memory

2.1.3 DESCRIPTION

When a wavefront is dispatched by the dispatcher, the base values, start address and EXEC mask values are generated and are sent as output from the fetch unit. WF_ID is generated for the new wavefront and the mapping between WF_TAG and WF_ID is stored in the register block. As the maximum number of wavefronts in a Compute Unit is 40, WF_ID can range from 0 to 40. Vacant register consists of 40 bits where each set bit corresponds to unassigned WF_ID. The PC block stores the next PC of instruction for all active wavefronts. The Round Robin scheduler schedules one of the dispatched wavefronts using the vacant register value. The WF_ID selected by scheduler is used to select BUFF_ADDR from the PC block. The selected PC is sent to the instruction memory and the incremented PC is stored back in the corresponding WF_ID slot in the PC block. The MSB of BUFF_TAG is set for the first instruction of the wavefront. PC and WF_ID of selected wavefront instruction are passed as a part of BUFF_TAG to the Instruction memory. The halt signal is received from the issue stage along with the corresponding WF_ID. The vacant bit for the WF_ID_DONE is set so that it is not scheduled again until another wavefront is assigned that WF_ID.

2.2 Wavegroup

2.3 Decode

The decode unit reads instructions fetched and decode them in fields. All fields decoded are described. All instructions, except halt, must have at least a 16 bit opcode, a identifier for the functional unit where it will be executed, a WF_ID and a PC value. SOMEBODY HAVE TO LIST WHAT EACH OPCODE DOES.

2.3.1 Functional units identifiers - FU_ID

Identifies functional units where instructions will be executed

00	Reserved value
01	Instruction will be executed by vALU
10	Instruction will be executed by sALU
11	Instruction will be executed by LSU

2.3.2 vALU opcode - VALU_OP

Identifies instructions that will be executed on vALU

VOP1 Instructions with this format have the following values on their opcode field:

VALU_OP[15:8] = 8'b0000_0001
VALU_OP[7:0] = VOP1_OP8

VOP2 Instructions with this format have the following values on their opcode field:

VALU_OP[15:8] = 8'b0000_0010
VALU_OP[5:0] = VOP2_OP6

2.3.3 sALU opcode - SALU_OP

Identifies instructions that will be executed on sALU

SOPP Instructions with this format have the following values on their opcode field:

SALU_OP[15:8] = 8'b0000_0001
SALU_OP[6:0] = SOPP_OP7

SOP1 Instructions with this format have the following values on their opcode field:

SALU_OP[15:8] = 8'b0000_0010
SALU_OP[7:0] = SOP1_OP8

SOP2 Instructions with this format have the following values on their opcode field:

SALU_OP[15:8] = 8'b0000_0100
SALU_OP[6:0] = SOP2_OP7

2.3.4 LSU opcode - LSU_OP

Identifies instructions that will be executed on LSU

SMRD Instructions with this format have the following values on their opcode field:

SMRD[15:8] = 8'b0000_0001

SMRD[5:0] = IMM,SMRD_OP5

MTBUFF Instructions with this format have the following values on their opcode field:

MTBUFF[15:8] = 8'b0000_0010

MTBUFF[7:0] = IDXEN,OFFEN,MTBUFF_OP3

2.3.5 Operand field encoding

The instructions opperands are encoded by OPERAND_REG_VALID and OPERAND_ID, with widths of 1 and 11 bits respectively.

OPERAND_REG_VALID: Flag that, when set, identifies whether the operand have to be read from vGPR or sGPR. If cleared the value is a state register or a constant.

OPERAND_ID: This value is encoded as follows:

If OPERAND_REG_VALID == 1'b1:
- If OPERAND_ID[10] == 1'b1:
  - If OPERAND_ID[9] == 1'b1.
    - OPERAND_ID[8:0] addresses one of 512 scalar registers in SGPR
  - If OPERAND_ID[9] == 1'b0:
    - OPERAND_ID[8:7] == 2'b0.
    - If OPERAND_ID[6:0] == 16'h01 the operand is VCC_LO
    - If OPERAND_ID[6:0] == 16'h02 the operand is VCC_HI
    - If OPERAND_ID[6:0] == 16'h04 the operand is VCC_Z. This encoding shall not be used for destination regs.
    - If OPERAND_ID[6:0] == 16'h08 the operand is EXEC_LO
    - If OPERAND_ID[6:0] == 16'h10 the operand is EXEC_HI
    - If OPERAND_ID[6:0] == 16'h20 the operand is EXEC_Z. This encoding shall not be used for destination regs.
    - If OPERAND_ID[6:0] == 16'h40 the operand is SCC. This encoding shall not be used for destination regs.
- If OPERAND_ID[10] == 1'b0:
  - OPERAND_ID[9:0] addresses one of 1024 vector registers in VGPR

If OPERAND_REG_VALID == 0:
- OPERAND_ID[10:0]denotes a constant supplied by decoder.

2.4 Issue

This unit combines a scoreboard and a arbiter. The scoreboard keeps track of all register being written by inflight instructions (i.e. have these registers as destination). All instructions whose operands depend on such registers are delayed untill the dependency is cleared. RAW and WAW dependencies are considered.

The issue unit choses one wavefront to issue from based on a round robin policy, from all wavefronts that can be issued at a given cycle. A instruction cannot bypass the scoreboard. It must rest on it for at least on cycle.

Every cycle, the issue stage can issue one instruction to LSU and another to SIMDs and SALU. Issue stage also keeps track of

2.5 SIMD

2.6 Scalar ALU

2.7 Load store unit

2.7.1 INTERFACE

MEM_TAG: It consists of MEM_TYPE and MEM_WF_ID. MEM_TYPE is the MSB, the other bits are MEM_WF_ID

MEM_TYPE: If this bit is set, the instruction issued is a load, otherwise it is a store

MEM_WF_ID: WF_ID of the wavefront that issued the memory operation

MEM_SGPR_SOURCE1 : Memory resource constant address

MEM_SGPR_SOURCE2 : SGPR (soffset)

MEM_VGPR_SOURCE1: VDATA VGPR

MEM_VGPR_SOURCE2: VGPR (voffset if OFFEN = 1, index if IDXEN = 1)

MEM_DEST_REG: Destination register for Load

MEM_INSTR_PC: PC of wavefront instruction

LOAD_VALUE: Load value from data memory

DEST_SGPR: Destination SGPR address

DEST_SGPR_WR: Destination SGPR write enable

DEST_VGPR: Destination VGPR address

DEST_VGPR_WR: Destination VGPR write enable

MEM_DONE: Set when ACK is received

MEM_DONE_WFID: WFID of completed memory instruction

2.7.2 DESCRIPTION

LSU supports two instruction types:

SMRD: Scalar Load

MTBUF: Vector Load or Store

The load store unit consists of 40 entry buffer addressed by WF_ID. Only one LD/ST instruction for a wavefront can be in flight. The issue stage keeps track of the issued Load-Store instruction for a particular WF_ID. Only after a Load-Store instruction in LSU buffer retires for a WF_ID, issue stage issues another LD/ST instruction. A valid bit for each entry indicates a valid entry for a WF_ID. The valid bit is cleared when a request is sent to memory. MEM_RD and MEM_WR are generated according to the MEM_TYPE bit of the buffer entry. For a non empty buffer, a request is issued to memory every cycle with priority given to lowest WF_ID. The ACK received from memory is out of order; therefore Issued_Instr buffer stores all the LD/ST requests stored in buffer. For store instructions, EXEC_MASK is also sent to memory. WF_ID(tag) is received along with ACK when a memory request is complete which is used to locate the corresponding MASK value, DEST_REG address and LOAD VALUE. The LD/ST instructions read from and write to SGPR and VGPR.

A Load store buffer entry consists of:
[MEM_ADDR, MEM_TYPE, STORE_VALUE, VALID]

An Issued_Instr entry consists of:
[LOAD_DEST_REG, STORE_VALUE_REG, INSTR_PC, EXEC_MASK, MEM_TYPE]