FUNCTIONAL SPECIFICATION FOR DOLPHIN ARITHMETIC PROCESSING ACCELERATOR

DONALD LEWINE

10-OCTOBER-78

ABSTRACT

This document describes the high speed arithmetic option for the DOLPHIN system. This option called the APA will speed up basic instructions as well as provide advanced functions such as square root, SIN, EXP etc.

This document also describes how the APA connects to the other elements of the DOLPHIN system.

1.0 OVERALL DESCRIPTION

The DOLPHIN APA monitors the IBOX and EBOX and speeds up selected operations. These include single and double precision floating point add, subtract, multiply and divide and single precision integer multiply and divide.

The following opcodes will be handled by the FPA:

Single Precision Floating Point
FAD, FADM, FADB, FADR, FADRI, FADRM, FADRB, FSB, FSBM, FSBB, FSBR, FSBRI, FSBRM, FMP, FMPM, FMPB, FMPR, FMPRI, FMPRM, FMPRB, FDV, FDVM, FDVB, FDVR, FDVRI, FDVRM, FDVRB

Single Precision Integer
MUL, MULI, MULM, MULB, IMUL, IMULI, IMULM, IMULB, DIV, DIVI, DIVM, DIVB, IDIV, IDIVI, IDIVM, IDIVB

Double Precision Floating Point
DFAD, DFSB, DFMP, DFDV.

Extended Range Double Precision Floating Point
EFAD, EFSB, EFMP, EFDV.

Conversion Instructions
FIX, FIXR, FLTR and new instructions for extended range conversions.

Basic FORTRAN Functions
SIN, COS, ATAN, SQRT, LOG, EXP, A*B+C, EMOD, POLY and maybe Dot Product. [These instructions have not been approved yet. We do not have any specific performance data to suggest how much these instructions would help FORTRAN programs. It is clear that they can help some programs. We need to find out how often this class of program occurs.]

If microcode space in the APA gets tight, the following instructions could be deleted at a fairly small performance loss:
The chopped instructions -- FADx, FSBx, FDVx, FMPx
The double AC integer instructions -- DIVx, MULx
Non-extended range double precision -- DFAD, DFSB, DFMP, DFDV

1.1 ACCURACY OF RESULTS

1.1.1 BASIC OPERATIONS - The APA functions which correspond to existing KL10 instructions, will give the same answers as the KS10 and KL10. In general, this answer is the same as one computed by the following procedure:

1. perform all computations in infinite precision keeping all bits of intermediate answers.

2. normalize the answer

3. round (or truncate) and store the answer.

The actual method used does not need infinite precision, however, the answer should match one produced by the above procedure.

1.1.2 COMPLEX OPERATIONS - THE PROCEDURE FOR COMPLEX OPERATIONS SHOULD BE THAT SAME AS FOR THE BASIC INSTRUCTIONS. ALL CONSTANTS AND INTERMEDIATE RESULTS NEED TO USE ENOUGH PRECISION SO THAT ADDING AN ARBITRARY AMOUNT OF ADDITIONAL PRECISION WILL NOT CHANGE THE ROUNDED RESULT. THE EXACT METHOD OF COMPUTING SIN, COS ETC. WILL REQUIRE CONSIDERABLE ANALYSIS.

2.0 CONNECTIONS TO REST OF DOLPHIN

The APA has its own copy of the ACs and can start an operation as soon as the IBOX fetch the instruction. The IBOX, EBOX, and APA may all be operating on separate instructions at any given time. The EBOX is required to help store the answer from the APA but does not have to start the APA or control any of the calculations.

The faster the APA can do arithmetic, the more the setup time hurts. In an effort to cut the setup time on frequent cases, the APA will try and start an instruction as soon as the opcode is known. If the instruction used indirection then, the APA may start an operation the indirect word instead of the data. There is no harm in this as long as the APA gets put on the right track after the effective address calculation is completed. The APA must be careful not to set any PC flags or store any answers until it is sure it is doing the right thing.

2.1 INPUTS AND OUTPUTS

The APA monitors the cache data lines from the MBOX. The APA can grab any data comming back from memory and start to operate on it. The APA can also start memory cycles for itself.

The APA also connects to the "write-bus". The APA can then keep its copy of the AC's up to date. It can also drive the bus to store in the AC's or memory. The APA will not store in memory on its own. The Ebox will always generate the write request and the APA and EBOX will work together to get the answer stored. this will assure all operations happen in the correct order.

The APA also monitors the IR and LAST IR registers in the IBOX. This lets the APA get started as soon as there is the possibility or an APA instruction. [[what about extends???]].

The APA drives 4 lines for setting PC flags. The arithmetic processing accelerator can set OVERFLOW, FLOATING OVERFLOW, FLOATING UNDERFLOW, and NO DIVIDE. The EBOX executes a special function to enable these APA outputs to set the PC flags.

3.Ø INTERNALS OF THE ARITHMETIC PROCESSING ACCELERATOR

The APA consists of the following functional blocks:

1. A 72 bit datapath for direct manipulation of double precision numbers. The datapath will allow several related operations to take place at once.

2. A full shifter to allow rapid normalization and shifting. The shifter should be able to normalize 75% of all results in 1 step and 98% in two steps.

3. A 13-bit datapath to allow exponent calculations on standard and extended range numbers.

4. A connection to the ACs and memory to allow the APA to fetch operands, store results and monitor the instruction stream.

5. There are 2 copies of the ACs to allow AC and AC+1 to be read in one operation. These RAMS also hold 128 double precision constants. [[We need to look at the FORTRAN math library and see if 128 is enough. We can add 256 more words if we have to.]]

6. There is a separate AR register on the ALU chip. This is included so that a 4-bit multiply step can take place in one 16.66 ns clock tick.

7. A micro controler to cycle the APA through various states. The APA micro controler will be very similar to the control in the EBOX. This will allow many of the EBOX MCA designs to be used in the APA.

A block diagram of the datapath is attached.

4.Ø PERFORMANCE

The APA is being designed to win FORTRAN benchmarks. With the APA the DOLPHIN should be 3.4 X a KL1Ø-PV (most likely) on SP1111 and other highly compute bound FORTRAN programs. The APA is responsible of most of this speedup.

opcode	kl(1)	dolphin (2)	ratio
DFMP	4.1Ø	Ø.433	9.5
DFDV	8.6Ø	1.2ØØ	7.1
DFAD	2.Ø3	Ø.333	6.1

Programs which make heavy use of the built-in functions (SIN, EXP, etc.) may be sped up much more than average.

The APA should be able to do 4 bits of multiply in 16.66 ns (24Ø million multiply steps per second). Divide will take 66 ns to divide 4 bits.

It is expected that the DOLPHIN APA will fit on two extended-hex modules.

- - - - - - - - - - - - - - - -

(1) KL1Ø-PV (SERIAL # 1Ø31) WITH MICROCODE VERSION 157. ONE HUNDRED PERCENT CACHE HIT. MEASURED BY DFKFB (SPEEDY).

(2) BEST GUESS. ONE HUNDRED PERCENT CACHE HIT. APA GRIND TIME PLUS TEN TICKS. WE MAY NOT BE ABLE TO MEET ALL THESE TIMES.