DFPIM Component Description (Preliminary)

This document provides a description of every DFPIM component. It includes a listing of all inputs and outputs for both signals and configuration. Performance, size, and energy characteristics of the components are included as they are developed. The components are organized in alphabetical order by component name.

The following components are included in this document:

DLY	Delay element to synchronize paths
FADD	Floating point add and subtract
FCNVR	Floating point ⇔ integer conversions
FDIV	Floating point divide (might be just an inverse?)
FIFO	FIFO buffer
FMUL	Floating point multiply
IALU	Integer ALU (add, subtract, compare, AND, OR, XOR, shift, rotate)
IDIV	Integer divide
IMUL	Integer multiply
LD	Load
LUT	Logical 6-input look up table
MEMxxx	Scratch pad memory 0.5KB, 4KB, 32KB, 256KB and 2048KB sizes
SLCT	Selects between 2 paths (mux, possible implement in switch network)
ST	Store
STMAC	State machine sequencer

DLY

The DLY component equalizes the path lengths of DFG branches that have different intrinsic path delays. The DLY element is added to shorter paths making them the same length as the longest path.

INPUTS

in_0	data input (8, 16, 32, or 64 bits)
avl_0	input data is available (1 bit)
clock	advances data on rising edge of the input clock (1 bit)
reset	clears internal available/ready bits (1 bit)

OUTPUTS

data	output data (8, 16, 32, or 64 bits)
ready	output data is ready for use (1 bit)

CONFIGURATION

size	select 8, 16, 32, or 64-bit data width (2 bits)
delay	delay data by 1, 2, 3, or 4 clocks (2 bits)

CHARACTERISTICS

Technology	40 nm	28 nm	20 nm	14 nm	
Size					
Energy					
Max clock					
Asynch delay					

FADD

The FADD component performs floating point addition, subtraction, and comparison. All operations are 64-bit IEEE-754 format with round to nearest. The component is fully pipelined with a (3 [hopefully] or 4?) clock latency. [Should there be a 32-bit option?]

INPUTS

in_0	input 0 data (64 bits)
avl_0	input 0 data is available (1 bit)
in_1	input 1 data (64 bits)
avl_1	input 1 data is available (1 bit)
clock	advances data on rising edge of the input clock (1 bit)
reset	clears internal available/ready bits (1 bit)

OUTPUTS

data	output data (64 bits)
ready	output data is ready for use (1 bit)
flag	tested condition is true (1 bit)

CONFIGURATION

imm	use immediate data (1 bit)
imdata	immediate data value (64 bits)
funct	selects add or subtract (1 bit)

CHARACTERISTICS

Technology	40 nm	28 nm	20 nm	14 nm	
Size					
Energy					
Max clock					
Asynch delay					

FCNVR

The FCNVR component converts between integer and floating point formats. All floating point values are 64-bit IEEE-754 format with round to nearest. The component is fully pipelined with a (3 [hopefully] or 4?) clock latency. Integer values are 64-bit twos complement [Should there be a 32-bit floating point option? Should there be 16, 32, 64-bit integer options? Should conversions be performed as part of a data-prep step rather than DFPIM operation?]

INPUTS

in_0	input 0 data (8, 16, 32, or 64 bits?)
avl_0	input 0 data is available (1 bit)
clock	advances data on rising edge of the input clock (1 bit)
reset	clears internal available/ready bits (1 bit)

OUTPUTS

data	output data (8, 16, 32, or 64 bits?)
flag	the tested condition is true (1 bit)
ready	output data is ready for use (1 bit)

CONFIGURATION

imm	use immediate data (1 bit)
imdata	immediate data value (64 bits)
size	select 8, 16, 32, or 64-bit data width (2 bits)
funct	selects from float-to-int and int-to-float operation (1 bit)

CHARACTERISTICS

Technology	40 nm	28 nm	20 nm	14 nm	
Size					
Energy					
Max clock					
Asynch delay					

FDIV or FINV?

The FDIV (or FINV) component performs a floating division (inverse). [An inverse operation can be followed by a multiplication to get a true division; N/D === N * 1/D. An inverse operation might save sufficient area and energy to make this worthwhile.]

INPUTS

in_0	input 0 data (8, 16, 32, or 64 bits)
avl_0	input 0 data is available (1 bit)
in_1	input 1 data (8, 16, 32, or 64 bits)
avl_1	input 1 data is available (1 bit)
clock	advances data on rising edge of the input clock (1 bit)
reset	clears internal available/ready bits (1 bit)

OUTPUTS

data	output data (8, 16, 32, or 64 bits)
flag	the tested condition is true (1 bit)
ready	output data is ready for use (1 bit)

CONFIGURATION

imm	use immediate data (1 bit)
imdata	immediate data value (64 bits)
size	select 8, 16, 32, or 64-bit data width (2 bits)
funct	selects xxxx (xx bit)
cond	tests for condition xxx (xx bit)
latency	input to output latency 0, 1, 2, or 3 clocks (2 bits)

CHARACTERISTICS

Technology	40 nm	28 nm	20 nm	14 nm	
Size					
Energy					
Max clock					
Asynch delay					

FIFO

The FIFO component provides asynchronous buffering between phases of a DFG implementation. The input and output width are separately configured for maximum flexibility.

INPUTS

wr_data	write data (8, 16, 32, or 64 bits)
wr_en	accept the write data (1 bit)
rd_en	remove read data from FIFO (1 bit)
avl_1	input 1 data is available (1 bit)
clock	advances data on rising edge of the input clock (1 bit)
reset	clears to empty FIFO (1 bit)

OUTPUTS

rd_data	output data from FIFO (8, 16, 32, or 64 bits)
empty	FIFO is empty (1 bit)
empty4	FIFO has less than 4 elements (1 bit)
full	FIFO is full (1 bit)
full4	FIFO has less than 4 spaces (1 bit)

CONFIGURATION

rd_size	select 8, 16, 32, or 64-bit data width output (2 bits)
wr_size	select 8, 16, 32, or 64-bit data width input (2 bits)

CHARACTERISTICS

Technology	40 nm	28 nm	20 nm	14 nm	
Size					
Energy					
Max clock					
Asynch delay					

FMUL

The FMUL component performs a floating point multiplication or multiply-accumulate. [Need to determine if accumulate adds too much complexity (size, energy, timing) or if separate FMUL/FADD is better overall approach?]

INPUTS

in_0	input 0 data (64 bits)
avl_0	input 0 data is available (1 bit)
in_1	input 1 data (64 bits)
avl_1	input 1 data is available (1 bit)
clock	advances data on rising edge of the input clock (1 bit)
reset	clears internal available/ready bits (1 bit)

OUTPUTS

data	output data (64 bits)
ready	output data is ready for use (1 bit)

CONFIGURATION

imm	use immediate data (1 bit)
imdata	immediate data value (64 bits)
//size	select 8, 16, 32, or 64-bit data width (2 bits)
funct	selects multiply or multiply-accumulat (1 bit)

CHARACTERISTICS

Technology	40 nm	28 nm	20 nm	14 nm	
Size					
Energy					
Max clock					
Asynch delay					

IALU

The IALU component performs integer operations on the two inputs. The operation is chosen from addition, subtraction, logical AND, OR, XOR, NOT, shift left, shift right arithmetic, shift right logical, rotate right, and rotate left. It also has a condition test function setting the flag output for the inputs being equal , not equal, less than, greater-equal, greater than, less-equal, equal 0, not equal 0, less than 0, greater-equal 0, greater 0, or less-equal 0. The IALU has a latency 0 option which provides a combinatorial output. This allows a single-cycle read-modify-write operation with a scratch pad memory. Only 1 level of combinatorial output is allowed at the specified DFPIM clock rate.

INPUTS

in_0	input 0 data (8, 16, 32, or 64 bits)
avl_0	input 0 data is available (1 bit)
in_1	input 1 data (8, 16, 32, or 64 bits)
avl_1	input 1 data is available (1 bit)
clock	advances data on rising edge of the input clock (1 bit)
reset	clears internal available/ready bits (1 bit)

OUTPUTS

data	output data (8, 16, 32, or 64 bits)
flag	the tested condition is true (1 bit)
ready	output data is ready for use (1 bit)

CONFIGURATION

imm	use immediate data (1 bit)
imdata	immediate data value (8, 16, 32, or 64 bits)
size	select 8, 16, 32, or 64-bit data width (2 bits)
funct	selects from the 11 available operations (4 bits)
cond	tests for 1 of the conditions (4 bits) [might have overlap]
latency	input to output latency 0, 1, 2, or 3 clocks (2 bits)

CHARACTERISTICS

Technology	40 nm	28 nm	20 nm	14 nm	
Size					
Energy					
Max clock					
Asynch delay					

IDIV

The IDIV component performs integer division or remainder …

INPUTS

in_0	input 0 data (8, 16, 32, or 64 bits)
avl_0	input 0 data is available (1 bit)
in_1	input 1 data (8, 16, 32, or 64 bits)
avl_1	input 1 data is available (1 bit)
clock	advances data on rising edge of the input clock (1 bit)
reset	clears internal available/ready bits (1 bit)

OUTPUTS

data	output data (8, 16, 32, or 64 bits)
flag	set for overflow (1 bit)
ready	output data is ready for use (1 bit)

CONFIGURATION

imm	use immediate data (1 bit)
imdata	immediate data value (64 bits)
size	select 8, 16, 32, or 64-bit data width (2 bits)
funct	selects between division and remainder (1 bit)
cond	tests for condition xxx (xx bit)
latency	input to output latency 1, 2, or 3 clocks (2 bits) [might need longer]

CHARACTERISTICS

Technology	40 nm	28 nm	20 nm	14 nm	
Size					
Energy					
Max clock					
Asynch delay					

IMUL

The IMUL component performs integer multiplication…

INPUTS

in_0	input 0 data (8, 16, 32, or 64 bits)
avl_0	input 0 data is available (1 bit)
in_1	input 1 data (8, 16, 32, or 64 bits)
avl_1	input 1 data is available (1 bit)
clock	advances data on rising edge of the input clock (1 bit)
reset	clears internal accumulator and available/ready bits (1 bit)

OUTPUTS

data	output data (8, 16, 32, or 64 bits)
flag	set for overflow (1 bit)
ready	output data is ready for use (1 bit)

CONFIGURATION

imm	use immediate data (1 bit)
imdata	immediate data value (64 bits)
size	select 8, 16, 32, or 64-bit data width (2 bits)
funct	selects multiply or multiply-accumulate (1 bit)
latency	input to output latency 0, 1, 2, or 3 clocks (2 bits)

CHARACTERISTICS

Technology	40 nm	28 nm	20 nm	14 nm	
Size					
Energy					
Max clock					
Asynch delay					

LD

The LD component is a small state machine optimized to read structured data from DRAM. It can access arrays of up to 3 dimensions. Configuration sets the base address, the bounds and stride of each dimension. It will fetch data elements consecutively or accept one or more indices from the dataflow fabric for address computation. [Perhaps an base-address/offset mode is needed as well? Direct address is then available with an offset of 0.]

More effort is needed to fully define internal options and verify operation through differt access methods. Should this component include ability to remap data into scratch pad memory? Such as convert a matrix from row-major to col-major order to ease matrix multiplications? Or should that be a different component?

[Might need an 'initiate' input to ensure it doesn't start fetching data before the DFG fabric is ready; or reset might supply that functionality.]

INPUTS

in_0	index 0 data (8, 16, or 32 bits)
avl_0	index 0 data is available (1 bit)
in_1	index 1 data (8, 16, or 32 bits)
avl_1	index 1 data is available (1 bit)
in_2	index 2 data (8, 16, or 32 bits)
avl_2	index 2 data is available (1 bit)
clock	advances data on rising edge of the input clock (1 bit)
reset	clears internal available/ready bits (1 bit)

OUTPUTS

data	output data (8, 16, 32, or 64 bits)
ready	output data is ready for use (1 bit)

CONFIGURATION

base_adrs	immediate data value (64 bits)
indx0bnd	index 0 bound (32-bits)
indx0strd	index 0 stride (32-bits)
indx1bnd	index 1 bound (32-bits)
indx1strd	index 1 stride (32-bits)
indx2bnd	index 2 bound (32-bits)
indx2strd	index 2 stride (32-bits)
size	select 8, 16, 32, or 64-bit data width (2 bits)

CHARACTERISTICS

Technology	40 nm	28 nm	20 nm	14 nm	
Size					
Energy					
Max clock					
Asynch delay					

LUT

The LUT component is a 6-input lookup table which allows the DFG fabric to make decisions. The flag outputs of the various components can be used to indicate when loop indices have reached their initial or terminal conditions and allow initialization or termination actions to be performed.

INPUTS

a	Boolean input a (1 bit)
b	Boolean input b (1 bit)
c	Boolean input c (1 bit)
d	Boolean input d (1 bit)
e	Boolean input e (1 bit)
f	Boolean input f (1 bit)
clock	advances data on rising edge of the input clock (1 bit)
reset	clears internal available/ready bits (1 bit)

OUTPUTS

g	Boolean output g (1 bit)
ready	output data is ready for use (1 bit)

CONFIGURATION

equation	table representing equation g = *f*(a,b,c,d,e,f) (64 bits)
latency	input to output latency 0 or 1 clock (1 bits)

CHARACTERISTICS

Technology	40 nm	28 nm	20 nm	14 nm	
Size					
Energy					
Max clock					
Asynch delay					

MEMxxx

The MEMxxx components are scratch pad memories of 512 bytes, 4 Kbytes, 32 Kbytes, 256 Kbytes, or 2048 Kbytes. [This is a first guess about useful sizes. The final size values will be determined after many more benchmarks have been analyzed to see what sizes and quantities are actually needed.] The memories can be configured as 1, 2, 4, or 8 bytes wide. There are 2 independent read ports and 2 independent write ports. It is up to the DFG fabric to prevent write collisions between the 2 write ports. The memories can be configured to have combinatorial or registered read, while writing is always synchronous to the rising edge of the clock input. [If the 3rd and 4th port result in a 'large' size, energy, or performance penalty, then dual-port memories, 1 read and 1 write, will be used.]

INPUTS

rd0_adrs	read port 0 address (log2(size) bits)
rd0_en	read port 0 enable (1 bit)
rd1_adrs	read port 1 address (log2(size) bits)
rd1_en	read port 1 enable (1 bit)
wr0_adrs	write port 0 address (log2(size) bits)
wr0_en	write port 0 enable (1 bit)
wr0_data	write port 0 data (8, 16, 32, or 64 bits)
wr1_adrs	write port 1 address (log2(size) bits)
wr1_en	write port 1 enable (1 bit)
wr1_data	write port 1 data (8, 16, 32, or 64 bits)
clock	advances data on rising edge of the input clock (1 bit)
reset	clears internal available/ready bits (1 bit) [not sure about a reset]

OUTPUTS

rd0_data	output data for port 0 (8, 16, 32, or 64 bits)
rd0_rdy	port 0 output data is ready for use (1 bit)
rd1_data	output data for port 1 (8, 16, 32, or 64 bits)
rd1_rdy	port 1 output data is ready for use (1 bit)

CONFIGURATION

size	select 8, 16, 32, or 64-bit data width (2 bits)
funct	selects xxxx (xx bit)
latency	input to output latency 0 or 1 clock (1 bits)

CHARACTERISTICS

Technology	40 nm	28 nm	20 nm	14 nm	
Size					
Energy					
Max clock					
Asynch delay					

SLCT

The SLCT component is used to select the result from alternate paths in the DFG. It is basically a multiplexor. [It might be more efficient to implement this within the DFG switching fabric.]

INPUTS

in_0	input 0 data (8, 16, 32, or 64 bits)
avl_0	input 0 data is available (1 bit)
in_1	input 1 data (8, 16, 32, or 64 bits)
avl_1	input 1 data is available (1 bit)
select	selector input, selects in_0 when '0' and in_1 when '1' (1-bit)
clock	advances data on rising edge of the input clock (1 bit)
reset	clears internal available/ready bits (1 bit)

OUTPUTS

data	output data (8, 16, 32, or 64 bits)
ready	output data is ready for use (1 bit)

CONFIGURATION

size	select 8, 16, 32, or 64-bit data width (2 bits)
latency	input to output latency 0, 1, 2, or 3 clocks (2 bits)

CHARACTERISTICS

Technology	40 nm	28 nm	20 nm	14 nm	
Size					
Energy					
Max clock					
Asynch delay					

ST

The ST component is a small state machine optimized to store structured data to DRAM. It can access arrays of up to 3 dimensions. Configuration sets the base address, the bounds and stride of each dimension. It will store data elements consecutively or accept one or more indices from the dataflow fabric for address computation. [Perhaps an base-address/offset mode is needed as well? Direct address is then available with an offset of 0.]

More effort is needed to fully define internal options and verify operation through differt access methods. Should this component include ability to remap data into scratch pad memory? Such as convert a matrix from row-major to col-major order to ease matrix multiplications? Or should that be a different component?

INPUTS

in_0	index 0 data (8, 16, or 32 bits)
avl_0	index 0 data is available (1 bit)
in_1	index 1 data (8, 16, or 32 bits)
avl_1	index 1 data is available (1 bit)
in_2	index 2 data (8, 16, or 32 bits)
avl_2	index 2 data is available (1 bit)
data	data to store (8, 16, 32, or 64 bits)
avail	data is ready for use (1 bit)
clock	advances data on rising edge of the input clock (1 bit)
reset	clears internal available/ready bits (1 bit)

OUTPUTS

CONFIGURATION

base_adrs	immediate data value (64 bits)
indx0bnd	index 0 bound (32-bits)
indx0strd	index 0 stride (32-bits)
indx1bnd	index 1 bound (32-bits)
indx1strd	index 1 stride (32-bits)
indx2bnd	index 2 bound (32-bits)
indx2strd	index 2 stride (32-bits)
size	select 8, 16, 32, or 64-bit data width (2 bits)

CHARACTERISTICS

Technology	40 nm	28 nm	20 nm	14 nm	
Size					
Energy					
Max clock					
Asynch delay					

STMAC

The STMAC component is a state machine sequencer component. Boolean signals are input and output. The input signals combined with the current_state determine the next_state determines the output signals.

INPUTS

a	input a data (1 bit)
avl_a	input a data is available (1 bit)
b	input b data (1 bit)
avl_b	input b data is available (1 bit)
c	input c data (1 bit)
avl_c	input c data is available (1 bit)
d	input d data (1 bit)
avl_d	input d data is available (1 bit)
e	input e data (1 bit)
avl_e	input e data is available (1 bit)
clock	advances data on rising edge of the input clock (1 bit)
reset	clears internal available/ready bits (1 bit)

OUTPUTS

w	output w data (1 bit)
x	output w data (1 bit)
y	output w data (1 bit)
z	output w data (1 bit)
stmac_rdy	output data is ready for use (1 bit)

CONFIGURATION

program	representation of state diagram (TBD bits)
in_msk	indicates which inputs are used (5 bits)

CHARACTERISTICS

Technology	40 nm	28 nm	20 nm	14 nm	
Size					
Energy					
Max clock					
Asynch delay					

TEMPLATE

The TEMPLATE component.

INPUTS

in_0	input 0 data (8, 16, 32, or 64 bits)
avl_0	input 0 data is available (1 bit)
in_1	input 1 data (8, 16, 32, or 64 bits)
avl_1	input 1 data is available (1 bit)
clock	advances data on rising edge of the input clock (1 bit)
reset	clears internal available/ready bits (1 bit)

OUTPUTS

data	output data (8, 16, 32, or 64 bits)
flag	the tested condition is true (1 bit)
ready	output data is ready for use (1 bit)

CONFIGURATION

imm	use immediate data (1 bit)
imdata	immediate data value (64 bits)
size	select 8, 16, 32, or 64-bit data width (2 bits)
funct	selects xxxx (xx bit)
cond	tests for condition xxx (xx bit)
latency	input to output latency 0, 1, 2, or 3 clocks (2 bits)

CHARACTERISTICS

Technology	40 nm	28 nm	20 nm	14 nm	
Size					
Energy					
Max clock					
Asynch delay					

TEMPLATE

The TEMPLATE component.

INPUTS

in_0	input 0 data (8, 16, 32, or 64 bits)
avl_0	input 0 data is available (1 bit)
in_1	input 1 data (8, 16, 32, or 64 bits)
avl_1	input 1 data is available (1 bit)
clock	advances data on rising edge of the input clock (1 bit)
reset	clears internal available/ready bits (1 bit)

OUTPUTS

data	output data (8, 16, 32, or 64 bits)
flag	the tested condition is true (1 bit)
ready	output data is ready for use (1 bit)

CONFIGURATION

imm	use immediate data (1 bit)
imdata	immediate data value (64 bits)
size	select 8, 16, 32, or 64-bit data width (2 bits)
funct	selects xxxx (xx bit)
cond	tests for condition xxx (xx bit)
latency	input to output latency 0, 1, 2, or 3 clocks (2 bits)

CHARACTERISTICS

Technology	40 nm	28 nm	20 nm	14 nm	
Size					
Energy					
Max clock					
Asynch delay					