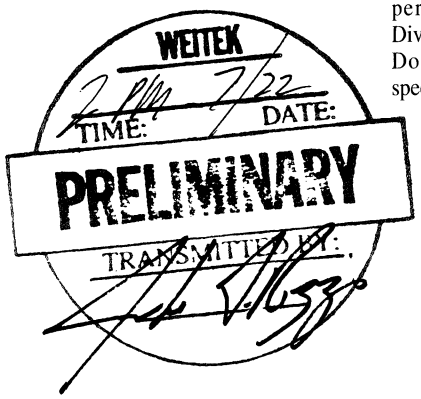
WEITEK

WTL 2264/WTL 2265 FLOATING POINT MULTIPLIER/DIVIDER AND ALU

PRELIMINARY DATA

July, 1986

The Weitek WTL 2264 Floating Point Multiplier and WTL 2265 Floating Point ALU provide high speed 32 and 64-bit numeric processing. Each chip can deliver up to 20 MFLOPs of single precision, and 12 MFLOPs of double precision performance. Single precision Divide occurs at a 500ns rate, with Double precision reaching 1μs speeds.



Contents

WTL 2264/WTL 2265
FLOATING POINT MULTIPLIER/ DIVIDER AND ALU

PRELIMINARY DATA
July 1986

Features

HIGH SPEED

20 MFlops (50 ns) pipelined for 32-bit ALU operations and 64-bit accumulations
20 MFlops (50 ns) pipelined for 32-bit multiplications
12 MFlops (80 ns) pipelined for 64-bit multiplications

FULL FUNCTION

Addition
Subtraction
Multiplication
Division
Conversion to and from two's complement integer
Compare
Absolute value
32-bit integer ALU operations

FULL INTERNAL 64-BIT ACCUMULATION PATH (WTL 2265)

CONFORMANCE TO IEEE STANDARD 754, VERSION 10.0

Full 32-bit and 64-bit floating point formats and operations

THREE 32-BIT PORTS

Two data inputs and one result output every 50 ns

LOW POWER CMOS

One Watt power dissipation

STANDARD 144-PIN PIN GRID ARRAY

Description

The WTL 2264 floating point multiplier/divider and the WTL 2265 floating point ALU provide high speed 32-bit and 64-bit floating point processing.

By virtue of their high I/O bandwidth and flexible pipeline structure, the WTL 2264/2265 can optimize either single precision or double precision performance. If optimization of double precision throughput is desired, the WTL 2264/2265-80 should be used. With the WTL 2264/2265-80, either single or double precision additions and multiplications can proceed at a 12.5 MFLOP (80 ns) rate. The latency for single precision operations is 280 ns, while the latency for double precision operations is 320 ns.

To optimize single precision performance or double precision latency, the WTL 2264/2265-50, -60 or -75 should be used. With the WTL 2264/2265-50, all single precision operations can be performed at a 20 MFLOP (50 ns) rate with a latency of 200 ns. Double precision multiplications can be performed with the 2264-50 at a 10 MFLOP rate with a latency of 300 ns. Double precision addition can also be performed at a 10 MFLOP rate, but has a latency of 250 ns.

In the "compatibility" mode, the WTL 2264/2265 are form-, fit- and function-compatible with the WTL 1264 and WTL 1265. Consequently, by asserting the compatibility mode bit, code written for the WTL 1264/1265 will run on the WTL 2264/2265. New routines may be written or speed critical routines re-written to take advantage of the WTL 2264/2265's higher throughput and lower latency. The higher throughput may be used to upgrade a system's performance.

This flexible two-chip set performs operations on single (32-bit) and double (64-bit) precision operands corresponding to IEEE Standard 754, Version 10.0 and 32-bit two's complement integers. Conformance to the standard includes all rounding modes, infinity and reserved operand representations, and the treatment of exceptions, such as overflow, underflow, invalid and inexact. Exact conformance also ensures complete software portability between systems designed using these devices and other general purpose computer systems which may be used to prototype algorithms and applications software. A "FAST" mode, which removes the time penalty of underflow exception handling by substituting zero for denormalized numbers, is included. All other features of the specification are retained in "FAST" mode.

Since the function specification is also pipelined there is no time penalty for interleaving various functions. Internal timers on the chips permit the pipeline to advance automatically so explicit pipeline flushing by pushing in new data is not required.

Description, continued

The WTL 2265 has a 64-bit internal accumulation path that allows the summation of 32- and 64-bit numbers without an external feedback path. Thus one cycle of latency can be saved on all accumulate operations, simplifying programming.

The devices' flexible I/O structure allows them to be integrated into systems with one, two or three 32-bit buses or one 64-bit bus. This flexibility provides an easy interface to systems with a variety of memory configurations. Separate input and output controls are used to program the ports. All inputs and outputs are fully registered and are loaded on each positive-going transition of the clock.

The function control determines the arithmetic operation to be performed. A 4-bit status output flags arithmetic exceptions and conditions. Function inputs and status outputs propagate with the corresponding data for designing and programming ease. A mode register selects optional characteristics that are not often changed, such as the selection of the IEEE rounding mode, which reduces microcode requirements.

Specifications

ABSOLUTE MAXIMUM RATINGS (Above Which The Useful Life May Be Impaired)

Supply voltage ..... −0.5 to 7.0 V
Input voltage ..... −0.5 to V_{DD}
Output voltage ..... −0.5 to V_{DD}
Operating temperature range
(T_{CASE}) ..... −55°C to 125°C
Storage temperature range ..... −65°C to 150°C
Lead temperature (10 seconds) ..... 300°C
Junction temperature ..... 175°C

RECOMMENDED OPERATING CONDITIONS

PARAMETER	COMMERCIAL			UNIT
	MIN	NOM	MAX	
Supply voltage, V_{CC}	4.75	5.0	5.25	V
Supply voltage, V_{DD}	4.75	5.0	5.25	V
Operating temperature, T_{CASE}	0		85	°C

Specifications, continued

DC ELECTRICAL CHARACTERISTICS, 1

PARAMETER	TEST CONDITIONS	COMMERCIAL MIN	COMMERCIAL MAX	UNIT
V_{IH} High-level input voltage	V_{CC}/V_{DD} = MAX	2.0		V
V_{IL} Low-level input voltage	V_{CC}/V_{DD} = MIN		0.8	V
V_{OH} High-level output voltage	V_{CC}/V_{DD} = MIN; I_{OH} = −1 mA	2.4		V
V_{OL} Low-level output voltage	V_{CC}/V_{DD} = MIN; I_{OL} = 4 mA		0.4	V
I_{IH} Input high current	V_{CC}/V_{DD} = MAX; V_{IH} = 5V		10	µA
I_{IL} Input low current	V_{CC}/V_{DD} = MAX; V_{IL} = 0V		10	µA
I_{OZL} 3 state leakage current low	V_{CC}/V_{DD} = MAX; V_{IL} = 0V		10	µA
I_{OZH} 3 state leakage current high	V_{CC}/V_{DD} = MAX; V_{IH} = 5V		10	µA
I_{SB} Standby supply current	All V_{IN} = 2.4 (High Z)			
I_{SB} Standby supply current	All V_{IN} = V_{DD} (High Z)			

Note 1: Tc = 0° C to 85° C; V_{CC} /V_{DD} = +4.75V min. to +5.25V max.

Specifications, continued

AC ELECTRICAL CHARACTERISTICS, 1

PARAMETER	TEST CONDITIONS (for all parameters)	2264/2265-75		2264/2265-60		2264/2265-50		UNIT
		MIN	MAX	MIN	MAX	MIN	MAX	
T_{CY} Clock cycle time	Note 1	75	DC	60	DC	50	DC	ns
T_{CH} Clock high time	Note 1	30						ns
T_{CL} Clock low time	Note 1	30						ns
T_S Input setup time	Note 1	15						ns
T_H Input hold time	Note 1	2						ns
T_D Output delay time	Note 1		35					ns
T_{VO} Output valid time	Notes 1 and 3	3						ns
T_{OZ} Output disable time	Note 1 modified by Note 2, Note 3		35					ns
T_{ZO} Output enable time	Note 1		35					ns
I_{CC} Supply current 2264	V_{DD}= MAX; TTL inputs T=0°C; T_{CY}= MIN		75					mA
I_{CC} Supply current 2265	V_{DD}= MAX; TTL inputs T=0°C; T_{CY}= MIN		75					mA
I_{DD} Supply current 2264	V_{DD}= MAX; TTL inputs T=0°C; T_{CY}= MIN		125					mA
I_{DD} Supply current 2265	V_{DD} = MAX; TTL inputs; T=0°C; T_{CY}= MIN		125					mA
T_{LA} Total Latency								
WTL 2265 32-bit operations & 64-bit accumulations			300		240		200	ns
WTL 2265 64-bit operations (2 new 64-bit inputs)			375		300		250	ns
WTL 2264 32-bit operations			300		240		200	ns
WTL 2264 64-bit operations			450		360		300	ns
WTL 2264 32-bit divisions			975		780		650	ns
WTL 2264 64-bit divisions			1,725		1,380		1,150	ns
T_{OP} Pipelined Time Per Stage								
WTL 2265 all operations			75		60		50	ns
WTL 2264 32-bit multiplications			75		60		50	ns
WTL 2264 64-bit multiplications			150		120		100	ns
WTL 2264 32-bit divisions			900		720		600	ns
WTL 2264 64-bit divisions			1,575		1,260		1,050	ns

Note 1: All switching characteristics are tested with inputs of 0.4 and 3.5V; output delay load shown in Figure 15. T_C = 0° C to 85° C; V_{CC} /V_{DD} = 4.75 min. to +5.25 max; timing transitions are are measured at 1.5V unless otherwise stated.
Note 2: Output load shown in Figure 16.
Note 3: Guaranteed but not tested.
Specifications subject to change without notice.

Specifications, continued

AC ELECTRICAL CHARACTERISTICS, 1

PARAMETER	TEST CONDITIONS (for all parameters)	2264/2265-80		2264/2265-100		UNIT
		MIN	MAX	MIN	MAX	
T_{CY} Clock cycle time	Note 1	40	DC	50	DC	ns
T_{CH} Clock high time	Note 1			20	DC	ns
T_{CL} Clock low time	Note 1			20	DC	ns
T_S Input setup time	Note 1			15		ns
T_H Input hold time	Note 1			2		ns
T_{VO} Output valid time	Notes 1 and 3			3		ns
T_D Output delay time	Note 1				35	ns
T_{OZ} Output disable time	Note 1 modified by Note 2; Note 3				35	ns
T_{ZO} Output enable time	Note 1				35	ns
I_{CC} Supply current 2264	V_{DD}= MAX; TTL inputs; T=0°C; T_{CY}= MIN				75	mA
I_{CC} Supply current 2265	V_{DD}= MAX; TTL inputs; T=0°C; T_{CY}= MIN				75	mA
I_{DD} Supply current 2264	V_{DD}= MAX; TTL inputs; T=0°C; T_{CY}= MIN				125	mA
I_{DD} Supply current 2265	V_{DD} = MAX; TTL inputs; T=0°C; T_{CY}= MIN				125	mA
T_{LA} Total Latency						
WTL 2265 32-bit operations & 64-bit accumulations			280		350	ns
WTL 2265 64-bit operations (2 new 64-bit inputs)			320		400	ns
WTL 2264 32-bit operations			280		350	ns
WTL 2264 64-bit operations			320		400	ns
WTL 2264 32-bit divisions			640		850	ns
WTL 2264 64-bit divisions			1,000		1,300	ns
T_{OP} Pipelined Time Per Stage						
WTL 2265 all operations			80		100	ns
WTL 2264 32-bit multiplications			80		100	ns
WTL 2264 64-bit multiplications			80		100	ns
WTL 2264 32-bit divisions			600		750	ns
WTL 2264 64-bit divisions			960		1,200	ns

Note 1: All switching characteristics are tested with inputs of 0.4 and 3.5V; output delay load shown in Figure 15. T_C = 0° C to 85° C; V_{CC} /V_{DD} = 4.75 min. to +5.25 max; timing transitions are are measured at 1.5V unless otherwise stated.
Note 2: Output load shown in Figure 16.
Note 3: Guaranteed but not tested.
Specifications subject to change without notice.

Block Diagram

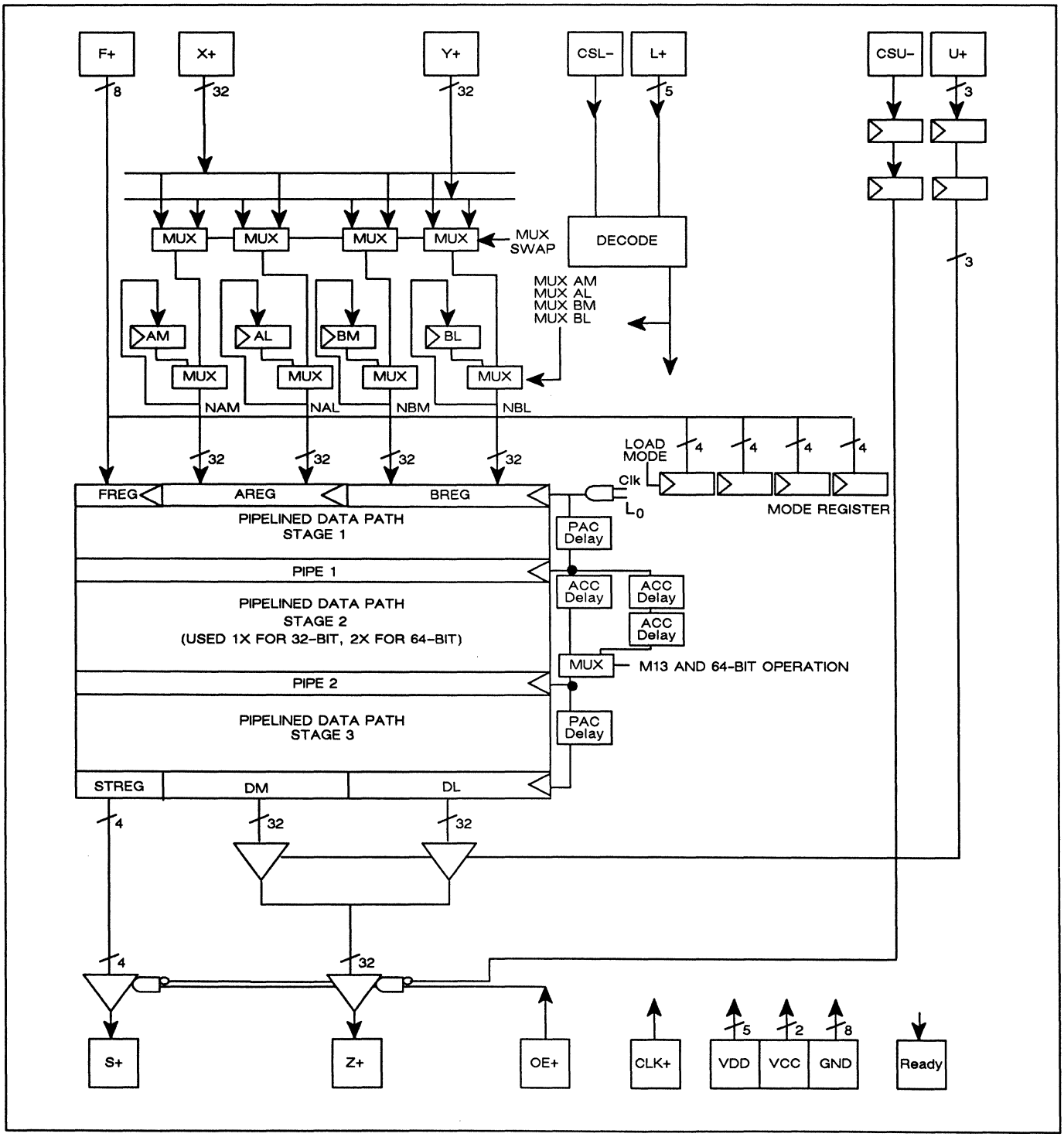


Figure 1. WTL 2264 Block Diagram

Block Diagram, continued

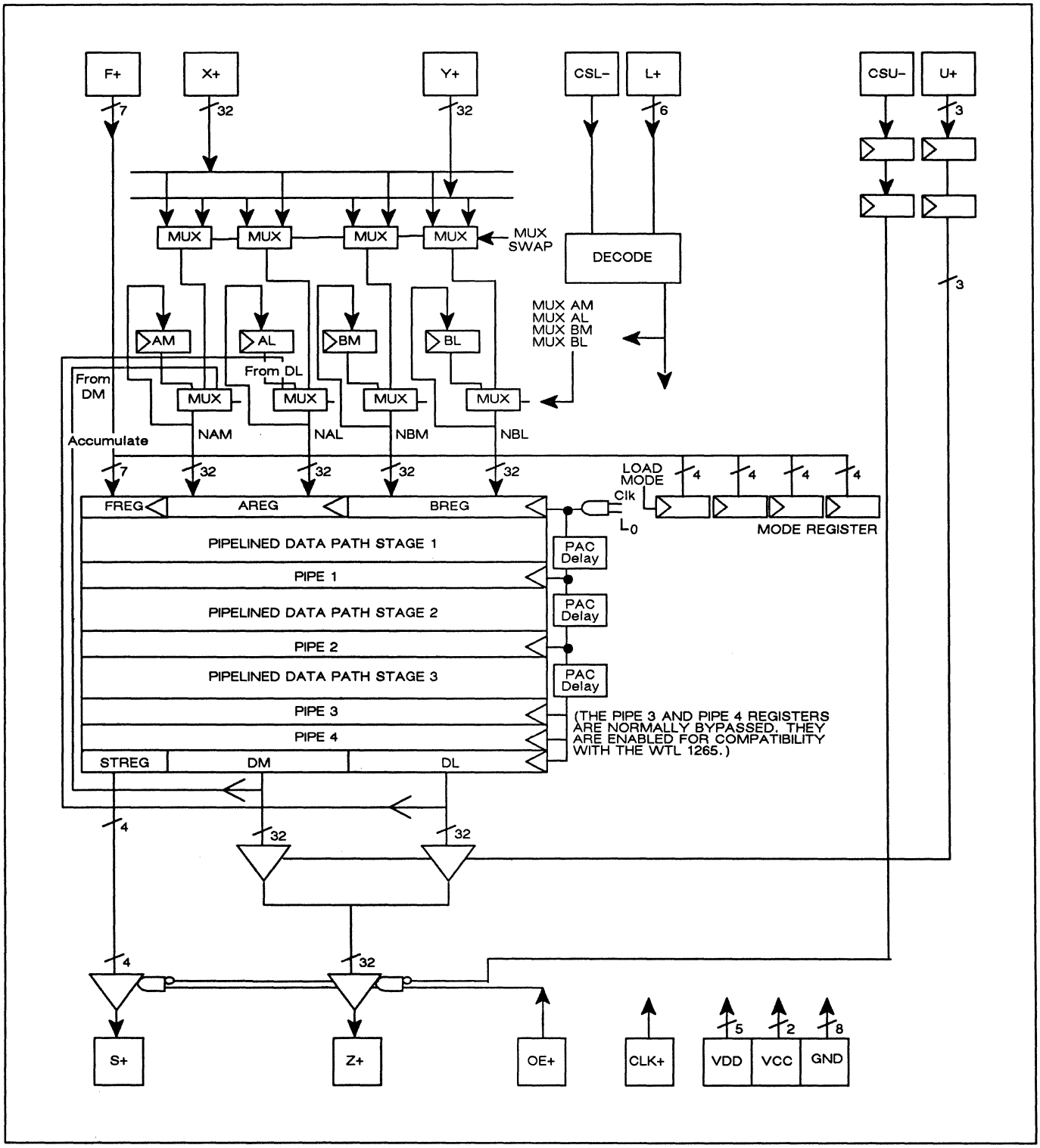


Figure 2. WTL 2265 Block Diagram

Signal Description

X_{31-0}

32-bit X Input Port

Y_{31-0}

32-bit Y Input Port

L_{5-0}

The 6-bit load control is used to implement the input configuration of the WTL 2264/2265 by specifying the destination (AM, AL, BM, BL or arithmetic array) of data on the X and Y inputs. The WTL 2264 uses only five bits of the load control; L_5 should be tied to ground on the WTL 2264.

F_{6-0}

7-bit function control on the WTL 2265; 8-bit function control for the WTL 2264. F_7 is unused on the WTL 2265 and should be tied to ground.

CSL–

A logic "0" on the CSL– pin enables the Load Control Bus; a logic "1" forces a Nop on the Load Control Bus. The CSL– signal allows the WTL 2264/2265 to share Data Input and Control buses.

Z_{31-0}

32-bit Z Output Port

S_{3-0}

Four-bit status output — indicates any exceptions or conditions that result from operations performed by the floating point units.

U_{2-0}

The unload control is used to specify the output source.

OE

A logic "0" on the asynchronous output enable *disables* outputs, while a logic "1" *enables* outputs.

CSU–

This signal is a synchronous output enable that allows the microcode to control the tri-stating of the output bus. This signal is staged with the same latency as the U_{2-0} field.

READY

For division, READY goes high for one cycle, on the cycle that partial results are clocked into Pipe Stage 3 of the WTL 2264. See figures 9 through 12 for more detail.

V_{DD}

The 5V power to the internal circuitry.

Vcc

The 5V power to the devices' output drivers

GND

Ground

CLK

Clock

Method of Operation

Both the WTL 2264 and the WTL 2265 consist of a pipelined arithmetic array with two input ports and one output port. Arithmetic operations are selected by the function control, while the input and output ports have 6-bit and 3-bit control fields. This section will discuss the hardware and controls of the WTL 2264/2265.

Method of Operation, continued

INPUT PORTS

The WTL 2264 and WTL 2265 have the same input structure and controls. Data presented on either of the 32-bit X or Y buses may be written into the on-chip registers (AL, AM, BL, BM) and/or may be passed directly into the arithmetic unit. Data input and output transfer may occur at twice the maximum double precision pipeline rate.

The large number of internal destinations for input data, combined with the high I/O bandwidth, permits the WTL 2264/2265 to be used in a variety of bus configurations for both 32- and 64-bit operations (one or two 32-bit input buses *or* one 64-bit input bus).

All inputs are fully registered, and, if CSL– is at logic "0", can be loaded on each positive-going transition of the clock. Transfers from the input ports to the internal registers, the ALU or the multiplier array are accomplished with the load controls $L_{5\text{-}0}$ as specified below. L_0 controls the initiation of an operation. If L_0 is a logic "0", only a data transfer occurs. If L_0 is a logic "1" the specified data is transferred and an operation begun as follows: the AREG and BREG are loaded with data from the specified AL, AM, BL or BM registers and the X and Y ports; the FREG is loaded from the F Port; and the operation specified by the FREG begins with the data loaded in the AREG and BREG.

Whenever L_5 is asserted on the WTL 2265, the contents of the DM/DL registers rather than the contents of the AM/AL registers are transferred to the AREG at the beginning of an operation. This allows on-chip accumulation.

Both X and Y ports can be used for unary operations. Unary operands (both integer and 32-bit floating point) must be loaded into the AREG. All 32-bit floating point operands must be loaded into the more significant half of the AREG or BREG. All 32-bit integer operands must be loaded into the less significant half of the AREG or BREG.

Load Controls

L_5 L_4 L_3 L_2 L_1 L_0	OPERATION
0 0 0 0 0 0	(nop)
0 0 0 0 0 1	AM,AL → AREG; BM, BL → BREG; F → FREG
0 0 0 0 1 0	Load mode
0 0 0 0 1 1	Reserved
0 0 0 1 0 0	Y → AL; X → BL
0 0 0 1 0 1	Y → AL; X → BL, AM,Y→ AREG;BM,X→ BREG; F→ FREG
0 0 0 1 1 0	Y → AM; X → BM
0 0 0 1 1 1	Y → AM; X → BM, Y, AL→ AREG;X,BL→ BREG; F→ FREG
0 0 1 0 0 0	X → BM; Y → BL
0 0 1 0 0 1	X → BM; Y → BL, AM,AL→ AREG;X,Y→ BREG; F→ FREG
0 0 1 0 1 0	X → AM; Y → AL
0 0 1 0 1 1	X → AM; Y → AL, X,Y→ AREG;BM,BL→ BREG; F→ FREG
0 0 1 1 0 0	X → AL; Y → BL
0 0 1 1 0 1	X → AL; Y → BL, AM,X→ AREG;BM,Y→ BREG; F→ FREG
0 0 1 1 1 0	X → AM; Y → BM
0 0 1 1 1 1	X → AM; Y → BM,X,AL→ AREG;Y,BL→ BREG; F→ FREG
0 1 0 0 0 0	Y → BM
0 1 0 0 0 1	Y → BM; AM,AL→ AREG,Y,BL→ BREG; F→ FREG
0 1 0 0 1 0	Y → BL
0 1 0 0 1 1	Y → BL; AM,AL→ AREG,BM,Y→ BREG; F→ FREG
0 1 0 1 0 0	Y → AL
0 1 0 1 0 1	Y → AL; AM,Y→ AREG, BM,BL→ BREG; F→ FREG
0 1 0 1 1 0	Y → AM
0 1 0 1 1 1	Y → AM; Y,AL→ AREG, BM,BL→ BREG; F→ FREG

Method of Operation, continued

Load Controls, continued

L_5 L_4 L_3 L_2 L_1 L_0	OPERATION
0 1 1 0 0 0	X → BM
0 1 1 0 0 1	X → BM; AM,AL→ AREG,X,BL→ BREG; F→ FREG
0 1 1 0 1 0	X → BL
0 1 1 0 1 1	X → BL; AM,AL→ AREG,BM,X→ BREG; F→ FREG
0 1 1 1 0 0	X → AL
0 1 1 1 0 1	X → AL; AM,X→ AREG, BM,BL→ BREG; F→ FREG
0 1 1 1 1 0	X → AM
0 1 1 1 1 1	X → AM; X,AL→ AREG, BM,BL→ BREG; F→ FREG
1 X X X X X	* DM, DL → AREG rather than AM, AL (see next table)

* WTL 2265 only; L_5 should be grounded on the WTL 2264.

L_5 L_4 L_3 L_2 L_1 L_0	OPERATION		
1 0 0 0 0 0	(NOP)		
1 0 0 0 0 1	DM,DL → AM,AL,AREG	BM,BL → BREG	F → FREG
1 0 0 0 1 0	LOAD MODE		
1 0 0 0 1 1	RESERVED		
1 0 0 1 0 0	DM,DL → AM,AL	X → BL	
1 0 0 1 0 1	DM,DL → AM,AL,AREG	X → BL; BM,X → BREG	F → FREG
1 0 0 1 1 0	DM,DL → AM,AL	X → BM	
1 0 0 1 1 1	DM,DL → AM,AL,AREG	X → BM; X,BL → BREG	F → FREG
1 0 1 0 0 0	DM,DL → AM,AL	X → BM; Y → BL	
1 0 1 0 0 1	DM,DL → AM,AL,AREG	X → BM; Y → BL; X,Y → BREG	F → FREG
1 0 1 0 1 0	DM,DL → AM,AL		
1 0 1 0 1 1	DM,DL → AM,AL,AREG	BM,BL → BREG	F → FREG
1 0 1 1 0 0	DM,DL → AM,AL	Y → BL	
1 0 1 1 0 1	DM,DL → AM,AL,AREG	BM,Y → BREG; Y → BL	F → FREG
1 0 1 1 1 0	DM,DL → AM,AL	Y → BM	
1 0 1 1 1 1	DM,DL → AM,AL,AREG	Y → BM; Y,BL → BREG	F → FREG
1 1 0 0 0 0	DM,DL → AM,AL	Y → BM	
1 1 0 0 0 1	DM,DL → AM,AL,AREG	Y → BM; Y,BL → BREG	F → FREG
1 1 0 0 1 0	DM,DL → AM,AL	Y → BL	
1 1 0 0 1 1	DM,DL → AM,AL,AREG	Y → BL; BM,Y → BREG	F → FREG
1 1 0 1 0 0	INVALID		
1 1 0 1 0 1	DM,DL → AM,AL,AREG	BM,BL → BREG	F → FREG
1 1 0 1 1 0	INVALID		
1 1 0 1 1 1	DM,DL → AM,AL,AREG	BM,BL → BREG	F → FREG
1 1 1 0 0 0	DM,DL → AM,AL	X → BM	
1 1 1 0 0 1	DM,DL → AM,AL,AREG	X → BM; X,BL → BREG	F → FREG
1 1 1 0 1 0	DM,DL → AM,AL	X → BL	
1 1 1 0 1 1	DM,DL → AM,AL,AREG	X → BL; BM,X → BREG	F → FREG
1 1 1 1 0 0	INVALID		
1 1 1 1 0 1	DM,DL → AM,AL,AREG	BM,BL → BREG	F → FREG
1 1 1 1 1 0	INVALID		
1 1 1 1 1 1	DM,DL → AM,AL,AREG	BM,BL → BREG	F → FREG

Method of Operation, continued

These tables can be condensed by noting the following regularities. The information on nodes NAM, NAL, NBM and NBL (as shown in Figure 2) is controlled by L_1, L_2, L_3 and L_4, respectively, as shown in the table below. Node NAM feeds into the AM Register, while NAL → AL, NBM → BM and NBL → BL. If L_0 is asserted high, then the information on NAM and NAL is latched into the AREG, the information on NBM and NBL is latched into the BREG and the operation begins. If L_5 is asserted high, the control of NAM and NAL by L_1 through L_4 is superseded and NAM and NAL receive the contents of the DM and DL registers, respectively.

L_4 L_3 L_2 L_1	OPERATION
0 0 0 0	Nop
0 0 0 1	Load Mode
0 0 1 0	Y → NAL; X → NBL
0 0 1 1	Y → NAM; X → NBM
0 1 0 0	Y → NBL; X → NBM
0 1 0 1	Y → NAL; X → NAM
0 1 1 0	Y → NBL; X → NAL
0 1 1 1	Y → NBM; X → NAM
1 0 0 0	Y → NBM;
1 0 0 1	Y → NBL;
1 0 1 0	Y → NAL;
1 0 1 1	Y → NAM;
1 1 0 0	X → NBM;
1 1 0 1	X → NBL;
1 1 1 0	X → NAL;
1 1 1 1	X → NAM;

Load Sequences For Example Configurations

32-bit Operations With Two 32-bit Ports

Operation	L_4	L_3	L_2	L_1	L_0
Y,AL → AREG; X, BL → BREG; F → FREG; Y → AM; X → BM	0	0	1	1	1

64-bit Operations Using The X And Y Ports As A Single Port

Operation	L_4	L_3	L_2	L_1	L_0
X → AM; Y → AL	0	1	0	1	0
AM,AL → AREG; X,Y → BREG	0	1	0	0	1
F → FREG; X → BM; Y → BL					

Method of Operation, continued

UNLOAD CONTROLS

CSU–	U_2 U_1 U_0	OPERATION
0	0 0 0	DM_{31-0} →ZPORT; STREG → SPORT
0	0 0 1	DM_{15-0},DM_{31-16} → ZPORT; STREG → SPORT
0	0 1 0	DM_{31-16}, DL_{31-16}→ZPORT; STREG → SPORT
0	0 1 1	DL_{31-16},DM_{31-16} → ZPORT; STREG → SPORT
0	1 0 0	DL_{31-0} → ZPORT; STREG → SPORT
0	1 0 1	DL_{15-0} ,DL_{31-16} → ZPORT; STREG → SPORT
0	1 1 0	DM_{15-0}, DL_{15-0}, → ZPORT;STREG → SPORT
0	1 1 1	DL_{15-0},DM_{15-0} → ZPORT; STREG → SPORT
1	x x x	SPORT and ZPORT tri-stated

The results of 32-bit floating point operations are stored in the DM Register. The more significant half of the result of a 64-bit operation is stored in the DM Register and the less significant 32 bits are stored in the DL Register. The results of integer operations are stored in the DL Register. One 32-bit result is transferred from the DM or DL Register to the Z Port every cycle. The unload control determines what portions of the DM or DL register are transferred to the Z Register.

MULTIPLIER/DIVIDER

The multiplier in the WTL 2264 has five basic elements: front end circuits to detect exceptions, a fixed point multiplier array, a circuit in parallel with the multiplier for adding exponents, a shifter to normalize the result of the fixed point multiplication and an IEEE rounding circuit. Exceptions (NaN and denormalized input) are detected at the input of the WTL 2264 by the exception circuitry. The timing for an exception is the same as that for a normal multiplication. For valid inputs, the exponents are added by the exponent adder.

A synchronous multiplier array performs the mantissa multiplication. One pass is required for a single precision IEEE multiply, while two passes are required for a double precision multiplication. The number of cycles required to pass through the array depends upon the system cycle time and the WTL 2264 speed grade. If the system has a 50 ns cycle time and a WTL 2264-50 is used, one cycle is required to pass through the array. If, on the other hand, the system cycle time is 40 ns and a WTL 2264-50 is used, two cycles would be required to pass through the array. A programmable timer, called the accumulate timer, is used to determine the time required to make a pass through the array and indirectly determines when results are latched into the WTL 2264's Pipe 2 Register.

Renormalization and IEEE rounding are performed between the pipeline register and the DM/DL registers. Mode control bits M_{3-2} are used to select the desired rounding mode.

The pipeline registers, the DM Register and the DL Register can be made transparent by mode bits M_1, M_5 and M_4. At the beginning of every operation the AREG and BREG are clocked. This clock pulse will ripple down to the Pipe 1, Pipe 2 and DM/DL registers after delays that are specified by the Pipeline Advance Control, M_{9-8}, and the Pipe 2 Advance Control, M_{7-6}, and M_{13}. The delay between the AREG, BREG and Pipe Register 1, as well as the delay between Pipe Register 2 and the DM/DL registers, is specified by the Pipeline Advance Control. The delay between Pipe 1 and Pipe 2 is specified by the Pipe 2 Advance Control.

Method of Operation, continued

In order to achieve the highest possible performance, it was necessary to eliminate the direct multiplication and division of denormalized numbers in the WTL 2264. In "FAST" mode, denorms (DNRMs) at the input and unorms (UNRMs) at the output of both the WTL 2264 and WTL 2265 are treated as zero. In IEEE mode, the WTL 2264 flags denorms. They must then be sent to the ALU to be wrapped. Wrapped numbers (WNRMs) can then be multiplied or divided using the WTL 2264. The section on "IEEE COMPATIBILITY" discusses this in detail.

The floating point multiplier controls are given below.

Function Controls For Floating Point Multiplier

F_2 F_1 F_0	OPERATION (BREG op AREG)	DESCRIPTION
0 0 0 (0)	F32 op F32	Single operation
0 0 1 (1)	F64 op F64	Double operation
0 1 0 (2)	F32 op W32	Single operation, A wrapped
0 1 1 (3)	F64 op W64	Double operation, A wrapped
1 0 0 (4)	W32 op F32	Single operation, B wrapped
1 0 1 (5)	W64 op F64	Double operation, B wrapped
1 1 0 (6)	W32 op W32	Single operation, A & B wrapped
1 1 1 (7)	W64 op W64	Double operation, A & B wrapped

F_5 F_4 F_3	OPERATION (BREG op AREG)	DESCRIPTION
0 0 0 (0)	B op A	Operate
0 0 1 (1)	B op |A|	B operate magnitude of A
0 1 0 (2)	|B| op A	A operate magnitude of B
0 1 1 (3)	|B| op |A|	Magnitude of A operate B
1 0 0 (4)	-(B op A)	Operate and negate
1 0 1 (5)	B op (-|A|)	B operate negative value of A
1 1 0 (6)	(-|B|) op A	A operate negative value of B
1 1 1 (7)	-(|B| op |A|)	Negative value of A operate B

If F_6 is a logic "0", the operation is an IEEE multiply. If, on the other hand, F_6 is a logic "1", the function performed is an IEEE division with all the options shown above. F_7 is used to perform mixed-mode calculations. If F_7 is asserted high, then (F32 • F32 = F64) or (F32 ÷ F32 = F64) functions are performed. All combinations of sign specified by F_5, F_4 and F_3 are supported. Wrapped formats are not supported for mixed mode operations.

When denormalized operands are encountered in mixed-mode operations, the single precision operand must be converted to double precision and the operation executed in double precision.

Division takes the form of B ÷ A.

ALU

The WTL 2265's ALU consists of five basic elements: front end circuitry to detect exceptions, a shifter to denormalize the smaller of the two input operands, an adder, a shifter to renormalize results and circuits to perform the IEEE rounding. However, the user can consider it to be a simple ALU with two internal pipeline registers. Two additional pipeline registers are available to maintain compatibility with the WTL 1265. Any combination of the pipeline registers as well as the DM and DL registers can be made transparent by mode bits M_{7-4}. At the beginning of every operation the AREG and BREG are clocked. This clock pulse will ripple down to the Pipe 1, Pipe 2 and DM/DL registers after delays specified by the Pipeline Advance Control, M_{9-8}.

F_7 has no effect in the WTL 2265 and should be grounded.

The WTL 2265's function controls are given below.

Method of Operation, continued

Function Controls for ALU

F_6 F_5 F_4 F_3 F_2 F_1 F_0	Operation (AREG op BREG)	Description
0 0 0 0 0 0 0 (0)	F32 − F32	single subtract
0 0 0 0 0 0 1 (1)	F64 − F64	double subtract
0 0 0 0 0 1 0 (2)	|F32 − F32|	single magnitude of difference
0 0 0 0 0 1 1 (3)	|F64 − F64|	double magnitude of difference
0 0 0 0 1 0 0 (4)	Reserved	
0 0 0 0 1 0 1 (5)	Reserved	
0 0 0 0 1 1 0 (6)	Reserved	
0 0 0 0 1 1 1 (7)	Reserved	
0 0 0 1 0 0 0 (8)	−F32	single negate
0 0 0 1 0 0 1 (9)	−F64	double negate
0 0 0 1 0 1 0 (10)	Reserved	
0 0 0 1 0 1 1 (11)	Reserved	
0 0 0 1 1 0 0 (12)	Reserved	
0 0 0 1 1 0 1 (13)	Reserved	
0 0 0 1 1 1 0 (14)	Reserved	
0 0 0 1 1 1 1 (15)	Reserved	
0 0 1 0 0 0 0 (16)	F32 + F32	single add
0 0 1 0 0 0 1 (17)	F64 + F64	double add
0 0 1 0 0 1 0 (18)	|F32 + F32|	single magnitude of sum
0 0 1 0 0 1 1 (19)	|F64 + F64|	double magnitude of sum
0 0 1 0 1 0 0 (20)	|F32| + |F32|	single add magnitude
0 0 1 0 1 0 1 (21)	|F64| + |F64|	double add magnitude
0 0 1 0 1 1 0 (22)	Reserved	
0 0 1 0 1 1 1 (23)	Reserved	
0 0 1 1 0 0 0 (24)	F32	single identity
0 0 1 1 0 0 1 (25)	F64	double identity
0 0 1 1 0 1 0 (26)	Reserved	
0 0 1 1 0 1 1 (27)	Reserved	
0 0 1 1 1 0 0 (28)	|F32|	single absolute value
0 0 1 1 1 0 1 (29)	|F64|	double absolute value
0 0 1 1 1 1 0 (30)	Reserved	
0 0 1 1 1 1 1 (31)	Reserved	
0 1 0 0 0 0 0 (32)	Compare F32 − F32	single compare
0 1 0 0 0 0 1 (33)	Compare F64 − F64	double compare
0 1 0 0 0 1 0 (34)	Reserved	
0 1 0 0 0 1 1 (35)	Reserved	
0 1 0 0 1 0 0 (36)	Compare |F32| − |F32|	single compare magnitude
0 1 0 0 1 0 1 (37)	Compare |F64| − |F64|	double compare magnitude
0 1 0 0 1 1 0 (38)	Reserved	
0 1 0 0 1 1 1 (39)	Reserved	
0 1 0 1 0 0 0 (40)	Compare F32 − 0	single compare with zero
0 1 0 1 0 0 1 (41)	Compare F64 − 0	double compare with zero
0 1 0 1 0 1 0 (42)	Reserved	
0 1 0 1 0 1 1 (43)	Reserved	
0 1 0 1 1 0 0 (44)	Reserved	
0 1 0 1 1 0 1 (45)	Reserved	
0 1 0 1 1 1 0 (46)	Reserved	
0 1 0 1 1 1 1 (47)	Reserved	
0 1 1 0 0 0 0 (48)	U32 → D32 (Exact)	single unwrap exact value
0 1 1 0 0 0 1 (49)	U64 → D64 (Exact)	double unwrap exact value
0 1 1 0 0 1 0 (50)	D32 → W32	single wrap denormalized value
0 1 1 0 0 1 1 (51)	D64 → W64	double wrap denormalized value
0 1 1 0 1 0 0 (52)	U32 → D32 (Inexact)	single unwrap inexact value
0 1 1 0 1 0 1 (53)	U64 → D64 (Inexact)	double unwrap inexact vlaue
0 1 1 0 1 1 0 (54)	Reserved	
0 1 1 0 1 1 1 (55)	Reserved	
0 1 1 1 0 0 0 (56)	F32 → I32	single fix

Method of Operation, continued

Function Controls for ALU, continued

F_6 F_5 F_4 F_3 F_2 F_1 F_0	Operation (AREG op BREG)	Description
0 1 1 1 0 0 1 (57)	F64 → I32	double fix
0 1 1 1 0 1 0 (58)	I32 → F32	single float
0 1 1 1 0 1 1 (59)	I32 → F64	double float
0 1 1 1 1 0 0 (60)	F32 → F64	convert single to double
0 1 1 1 1 0 1 (61)	F64 → F32	convert double to single
0 1 1 1 1 1 0 (62)	Reserved	
0 1 1 1 1 1 1 (63)	Reserved	
1 0 0 0 0 0 0 (64)	I32 − I32	integer subtract (A−B)
1 0 0 1 0 1 0 (74)	−I32 + I32	integer subtract (B−A)
1 0 0 0 0 0 1 (65)	I32 − I32 − BR	*integer subtract with borrow (A−B)
1 0 0 1 0 1 1 (75)	−I32 + I32 − BR	*integer subtract with borrow (B−A)
1 0 0 1 0 0 0 (72)	−I32	integer negate
1 0 0 1 0 0 1 (73)	−I32 + CR	*negate plus carry
1 0 1 0 0 0 0 (80)	I32 + I32	integer add
1 0 1 0 0 0 1 (81)	I32 + I32 + CR	*integer add plus carry
1 0 1 1 0 0 0 (88)	I32	pass
1 0 1 1 0 0 1 (89)	I32 + CR	*pass plus carry
1 1 0 1 0 0 0 (104)	I32 ∩ I32	and
1 1 0 1 0 1 0 (106)	$\overline{I32}$	not A
1 1 0 1 1 0 0 (108)	I32 ∧ I32	exclusive or
1 1 0 1 1 1 0 (110)	I32 ∪ I32	or

*For long word integer operations, the carry or borrow is only valid for one pipeline delay.

MODE CONTROLS

Mode controls are not directly loaded from dedicated pins. Instead, the LOAD MODE instruction uses the function control bits F_{5-4} to determine which one of four 4-bit subsets of the mode word is loaded through function bits F_{3-0}.

F_6 F_5 F_4	EDGE #1
0 0 0 (0)	F_{3-0} → M_{3-0}
0 0 1 (1)	F_{3-0} → M_{7-4}
0 1 0 (2)	F_{3-0} → M_{11-8}
0 1 1 (3)	F_{3-0} → M_{15-12}

"FAST"/IEEE Format Mode Control

Mode bit M_0 controls the treatment of denormalized numbers.

M_0	DENORMALIZED NUMBER HANDLING	
0	IEEE single or double format; multiplier and ALU generate denormalized operand exceptions and produce UNRM values on underflow exceptions.	(IEEE mode)
1	IEEE single or double format; multiplier and ALU flush denormalized operands to zero and round underflow results to zero.	(FAST mode)

Method of Operation, continued

Rounding Mode Control

Mode bits M_{3-2} determine the IEEE rounding mode for all operations except SINGLE FIX and DOUBLE FIX.

M_3	M_2		SELECTED ROUNDING MODE
0	0	(0)	Round toward nearest value or even significand, if a tie
0	1	(1)	Round toward zero
1	0	(2)	Round toward positive infinity
1	1	(3)	Round toward negative infinity

Rounding Mode Control For FIX Operations (WTL 2265 only)

Mode bit M_1 controls the IEEE rounding mode for SINGLE FIX and DOUBLE FIX.

$MODE_1$	SELECTED ROUNDING MODE
0	Round according to default rounding mode ($MODE_{3-2}$)
1	Round toward zero, regardless of the default rounding mode

Pipeline Configuration Control

Mode bits M_7, M_6 and M_4 of the floating point ALU and mode bits M_1, M_5 and M_4 of the floating point multiplier/divider determine which of the pipeline registers are latched and which are transparent. Pipeline Registers 3 and 4 on the ALU cannot be independently controlled. These registers are enabled to match the latency of the WTL 1265 in the compatibility mode. If a mode bit is set to logic "0", the corresponding register is transparent; if it is a logic "1", the register is latched by the rising edge of the clock.

	PIPE 1	PIPE 2	DM, DL
2264 MODE BIT	M_1	M_5	M_4
2265 MODE BIT	M_7	M_6	M_4

Compatibility Mode Control

The WTL 2264/2265 may be programmed for microcode compatibility with the WTL 1264 and WTL 1265. The most common WTL 1264/1265 configuration is selected by setting M_{12} to zero. In this mode 32-bit throughput and 64-bit throughput are equal and 64-bit throughput is maximized. For more information consult the application note at the end of this data sheet.

Method of Operation, continued

Pipe 2 Advance Control (WTL 2264 only)

Mode bits M_{7-6} (ACC) and M_{13} control the timing of the partial product accumulator in the WTL 2264.

ACC		M_{13}	PIPE 2 ADVANCE	
M_7	M_6		Single Precision	Double Precision
0	0	0	clock	clock/2
0	1	0	clock/2	clock/4
1	0	0	clock/3	clock/6
1	1	0	clock/4	clock/8
0	0	1	clock/2	clock/2
0	1	1	clock/4	clock/4
1	0	1	clock/6	clock/6
1	1	1	clock/8	clock/8

If mode bit M_{13} is a logic "0", Pipe Register 2 is latched [1 • (M_{7-6} + 1)] after Pipe Register 1 for single precision operations and [2 • (M_{7-6} + 1)] for double precision operations. If M_{13} is a logic "1", then single precision operations have the same timing as double precision operations and double precision operations are latched [2 • (M_{7-6} + 1)] cycles after Pipe 1.

Pipeline Advance Control (PAC)

Mode bits M_{9-8} control when the pipeline registers are latched, following the beginning of an operation. The pipeline registers are clocked at the beginning of every operation and N+1 cycles after the beginning of every operation, where N is given by mode bits M_{9-8}. In the WTL 2264, pipeline stages 1 and 3 are controlled by the Pipeline Advance Control, while Pipeline Stage 2 is controlled by the accumulator advance control. In the WTL 2265, pipeline stages 1, 2 and 3 are controlled by the Pipeline Advance Control. Division has unique timing and is described in a separate section beginning on page 23.

PAC M_9 M_8	PIPELINE RATE
0 0	clock
0 1	clock/2
1 0	clock/3
1 1	clock/4

Method of Operation, continued

SUMMARY OF MODE CONTROLS

MODE BIT	WTL 2265	WTL 2264
M_0	IEEE/"FAST" mode selection	IEEE/"FAST" mode
M_1	Round mode for "FIX" selection	Pipe 1 enable
$M_{2,3}$	Round mode selection	Round mode selection
M_4	DM, DL pipe enable	DM, DL pipe enable
M_5	Pipe 3, Pipe 4 enable	Pipe 2 enable
M_6	Pipe 2 enable	ACC – bit 0
M_7	Pipe 1 enable	ACC – bit 1
M_8	Pipeline advance	PAC – bit 0
M_9	Pipeline advance	PAC – bit 1
M_{10}	—	—
M_{11}	—	—
M_{12}	Compatibility	Compatibility
M_{13}	—	Combined with M_{7-6} = Pipe 2 Advance Control
M_{14}	—	—
M_{15}	—	—

RESULTS STATUS

The S Bus indicates any exceptions or conditions that result from operations performed by the WTL 2264 and WTL 2265. For floating point comparison operations, the S Bus indicates the condition resulting from the comparison. For all other floating point operations, the S Bus indicates exception status.

$S_3 S_2 S_1 S_0$	Comparison Condition	Exception Status
0 0 0 0 (0)	Equal	Result = +0 or –0, exact
0 0 0 1 (1)	Less than	Result = +infinity or –infinity, exact
0 0 1 0 (2)	Greater than	Result finite and $\neq$ 0, exact
0 0 1 1 (3)		Result finite and $\neq$ 0, inexact
0 1 0 0 (4)		– not used
0 1 0 1 (5)		Overflow & inexact
0 1 1 0 (6)		Underflow
0 1 1 1 (7)		Underflow & inexact
1 0 0 0 (8)		Operand A is denormalized
1 0 0 1 (9)		Operand B is denormalized
1 0 1 0 (10)		Operands A & B are denormalized
1 0 1 1 (11)		Divide by zero
1 1 0 0 (12)		Operand A is NaN
1 1 0 1 (13)		Operand B is NaN
1 1 1 0 (14)		Operands A & B are NaN
1 1 1 1 (15)	Unordered	Invalid Operation

Under certain conditions, multiple exceptions can occur. These exceptions may be resolved with the following priority table, where higher priority will mask lower priority exceptions.

Method of Operation, continued

Priority	Exception
Highest	Operands A & B are NaN
	Operand A is NaN
	Operand B is NaN
	Invalid Operation
	Divide by zero
	Operands A & B are denormalized
	Operand A is denormalized
	Operand B is denormalized
	Underflow & inexact
	Underflow
	Overflow & Inexact
Lowest	Result is finite and $\neq 0$, inexact

For fixed point operations (WTL 2265 functions 64–77), the status codes are as follows.

S_3	S_2	S_1	S_0	Exception Status
sign	0	0	0	Zero
sign	0	1	0	Result is finite and $\neq 0$
sign	1	0	1	Overflow

Timing

The WTL 2264 and WTL 2265 can be optimized for high performance for either single or double precision operations. For double precision operations, the binding constraints are the I/O time and the multiplier's array time.

Therefore, to achieve the highest throughput for double precision operations the WTL 2264/2265-80 or -100 should be used. These parts have the fastest I/O and array times (40 ns and 50 ns, respectively).

To achieve the highest performance for single precision operations the time spent in each pipeline stage must be minimized. Since the pipe times are shortest for the WTL 2264/2265-75, WTL 2264/2265-50 and WTL 2264/2265-60, these devices should be used when low latency and single precision throughput are to be maintained.

64-BIT MAXIMUM PIPELINED THROUGHPUT (WTL 2264-80 and -100)

To maximize throughput for 64-bit multiplications, set ACC to 00, M_{13} to "1" and PAC to 01, and put Pipe 1, Pipe 2 and the DM/DL registers in the latched mode. Figure 3 shows the timing. As soon as the inputs and function code are loaded, the operation begins. At the end of Cycle 4, as shown, the operands are at the output of Pipe 1, which is just before the multiplier array. One cycle after that, the Accumulate Register latches partial results from the first pass through the multiplier array. At the end of Cycle 6, the unrounded results of the multiplication are latched into the Pipe 2 Register. Two cycles are required between Pipe 2 and the DM/DL registers. At the end of Cycle 8, the results are at the input of the DM/DL registers.

Timing, continued

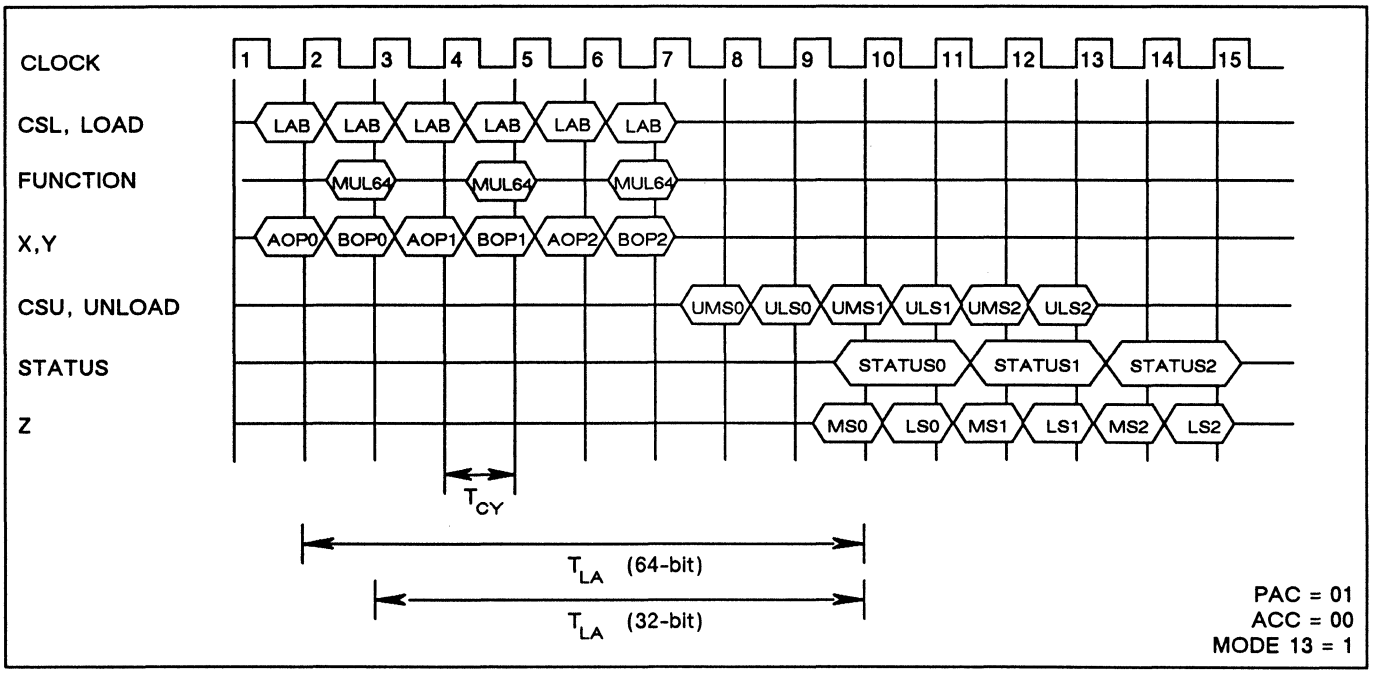


Figure 3. WTL 2264/2265-80 and -100 32- and 64-bit Pipelined Operation Timing

32-BIT MAXIMUM PIPELINED THROUGHPUT (WTL 2264-80 and -100)

Since the Pipeline Advance Control is set to 01, M_{13} is set to "1" and PAC is set to 00, single precision multiplications proceed at the same rate as double precision multiplications for the WTL 2264-80 and -100. Therefore, the timing illustrated in Figure 3 applies. The 32–bit operands may be loaded during cycle two to reduce the latency by one cycle.

32- AND 64-BIT ALU OPERATIONS (WTL 2265-80 and -100)

ALU operations have the same timing as multiplications for the WTL 2264/2265-80 and -100. Therefore, the timing shown in Figure 3 applies.

64-BIT MAXIMUM PIPELINED THROUGHPUT (WTL 2264-50, -60 or -75)

To maximize throughput for 64-bit multiplications, set ACC to 00, M_{13} to "0" and PAC to 00, and put Pipe 1, Pipe 2 and the DM/DL registers in the latched mode. The timing is described in Figure 4. As soon as the inputs and function code are loaded, the operation begins. At the end of Cycle 3, as shown, the operands are at the output of Pipe 1, just before the multiplier array. One cycle after that, the Accumulate Register latches partial results from the first pass through the multiplier array. At the end of Cycle 5, the unrounded results of the multiplication are latched into the Pipe 2 Register. At the end of Cycle 6, the results are received at the input of the DM/DL registers.

Timing, continued

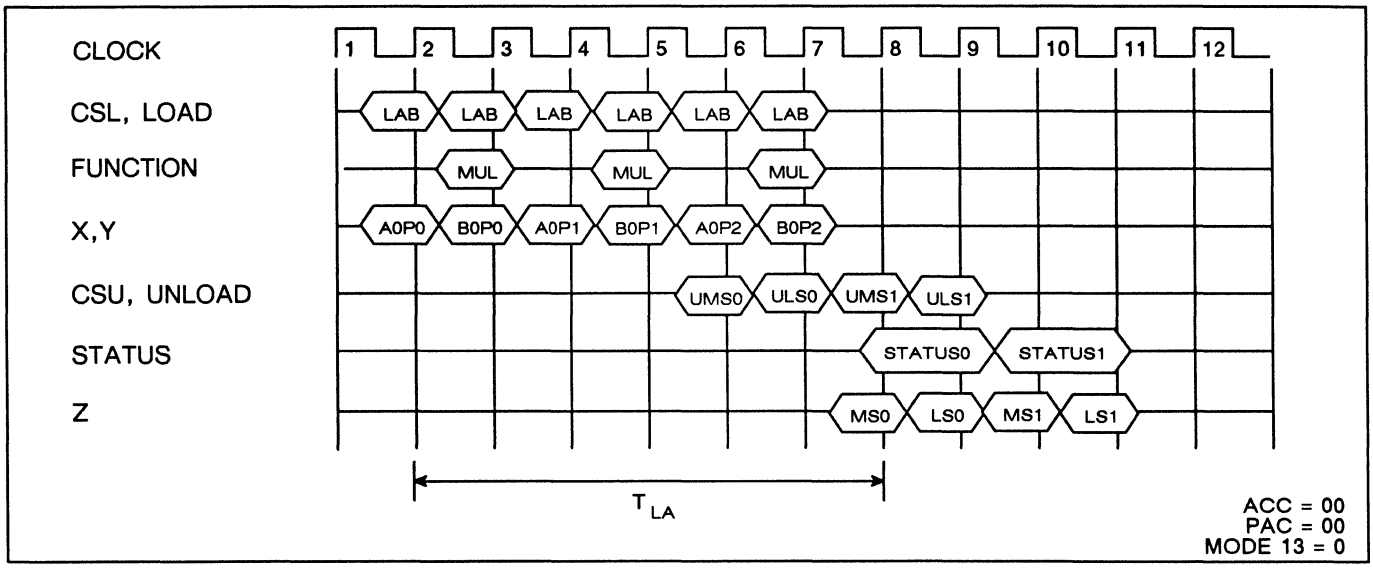


Figure 4. WTL 2264-50, -60 or -75 64-bit Pipelined Operation Timing

32-BIT MAXIMUM PIPELINED THROUGHPUT (WTL 2264-50, -60 or -75)

To obtain maximum throughput for 32-bit multiplications, set ACC to 00, M_{13} to "0" and PAC to 00, and put the pipeline registers and the DM/DL registers in the latched mode. The result will be located in the most significant half of the DM Register. The timing for 32-bit pipelined multiplications is shown in Figure 5. Note that the time spent in Pipe Stage 2 is one cycle for 32-bit operations and two cycles for 64-bit operations.

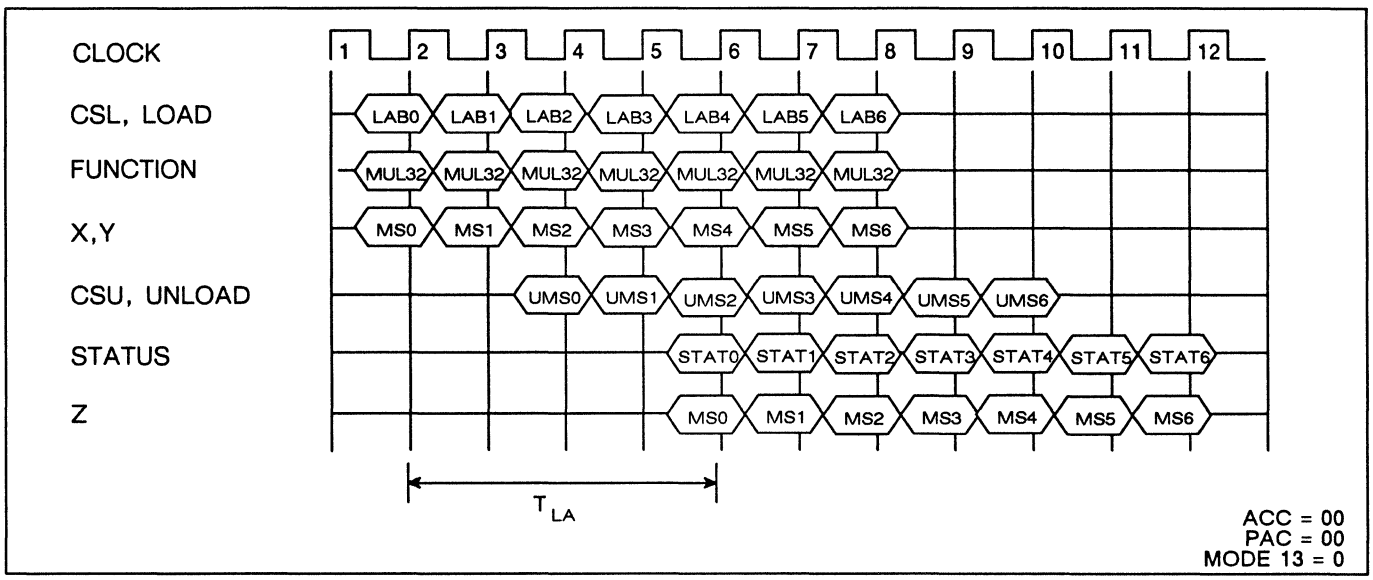


Figure 5. WTL 2264-50, -60 or -75 32-bit Pipelined Operation Timing

Timing, continued

MAXIMUM THROUGHPUT WITH INTERNAL ACCUMULATION (WTL 2265-50, -60 or -75)

The ALU has the same throughput for both 32- and 64-bit accumulations. For all other 64-bit operations, the WTL 2265 is I/O limited. To obtain maximum throughput, set PAC to 00 and activate all pipeline stages (except Pipe 3 and Pipe 4). The timing for pipelined throughput in the ALU is shown in Figure 6.

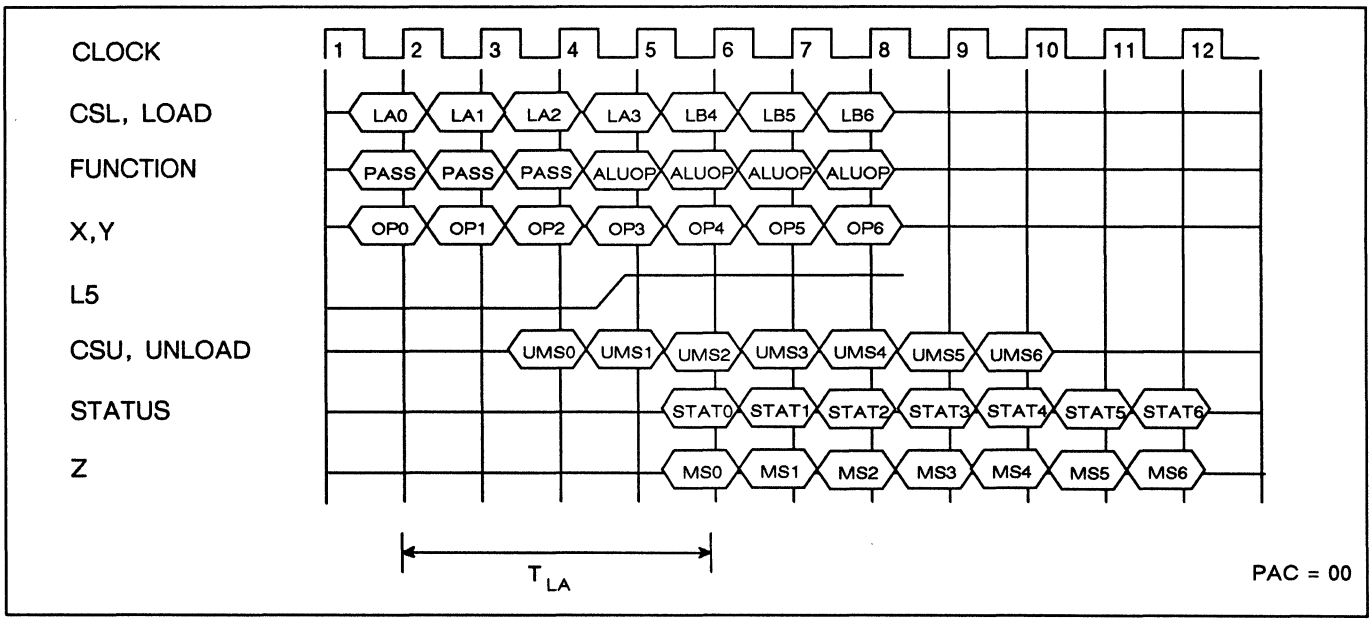


Figure 6. WTL 2265-50, -60 or -75 Accumulate Timing (when accumulating 32- or 64-bit numbers)

Timing, continued

MAXIMUM THROUGHPUT WITH INTERNAL ACCUMULATION (WTL 2265-80 or -100)

The timing for accumulate operations using the WTL 2265-80 and -100 speed devices is described in Figure 7. Set PAC to 01.

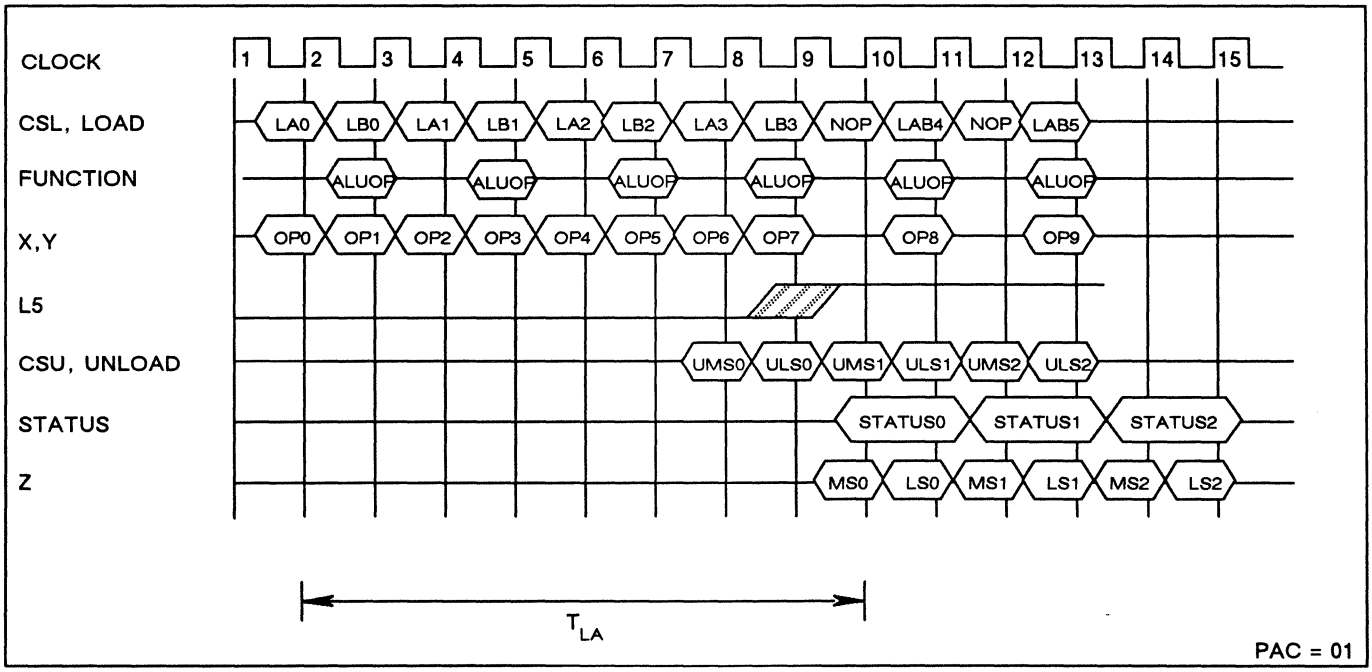


Figure 7. WTL 2265-80 or -100 Accumulate Timing

Timing, continued

MAXIMUM THROUGHPUT FOR ALU OPERATIONS WITH TWO NEW INPUTS (WTL 2265-50, -60 or -75)

For 32-bit ALU operations, the WTL 2265-50, -60 and -75 have the same throughput as 32-bit accumulations. For 64-bit ALU operations, on the other hand, two input cycles are required. Only one cycle per pipeline stage is required, however. To obtain maximum pipelined throughput, set PAC to 00 and activate all pipeline stages except Pipe 3 and Pipe 4. The timing is shown on the next page.

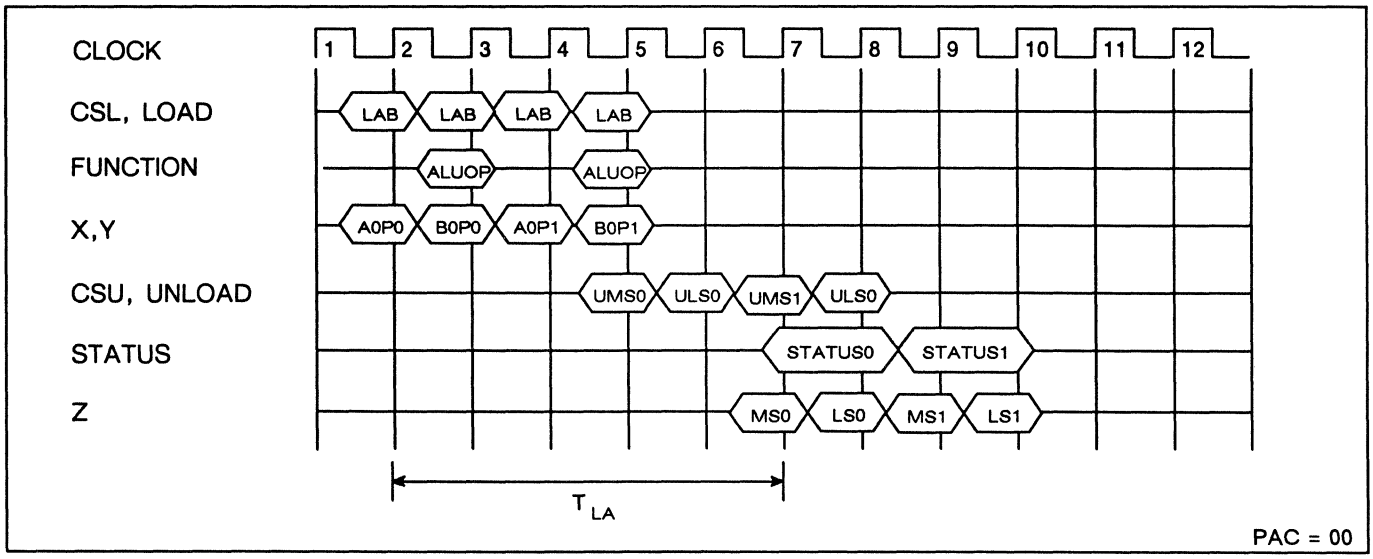


Figure 8. WTL 2265-50, -60 or -75 Pipelined ALU Operation Timing

DIVISION

Single precision division (F32 ÷ F32) has an operation time of (1 • PAC) + (9 • ACC) + (2 • PAC). The total latency is one cycle longer than this. Double precision division (F64 ÷ F64) has an operation time of (1 • PAC) + (18 • ACC) + (2 • PAC) and a latency that is two cycles longer than this. The values in parentheses designate the number of cycles that are spent, respectively, in pipe stages 1, 2 and 3 (WTL 2264). For division, M_{13} has no effect. The pipe 2 advance is always (9 • ACC) for single precision operations and (18 • ACC) for double precision operations.

A "normally low" READY signal is provided, which goes high for one cycle on the cycle in which partial results are clocked into pipe stage 3. For the WTL 2264-50, -60 and -75 another operation can begin two cycles later. For the WTL 2264-80 and -100 another operation can begin four cycles later. To ensure compatability with future upgrades to the WTL 2264/2265 the operation after a divide should be keyed by the READY signal.

If another function is loaded at any time during a division operation the second operation completes while the divide is corrupted.

Since the WTL 2264-80 and -100 require two cycles per pipeline stage, they have different timing, as is shown in the diagrams on the next two pages.

Division takes the form B ÷ A.

Timing, continued

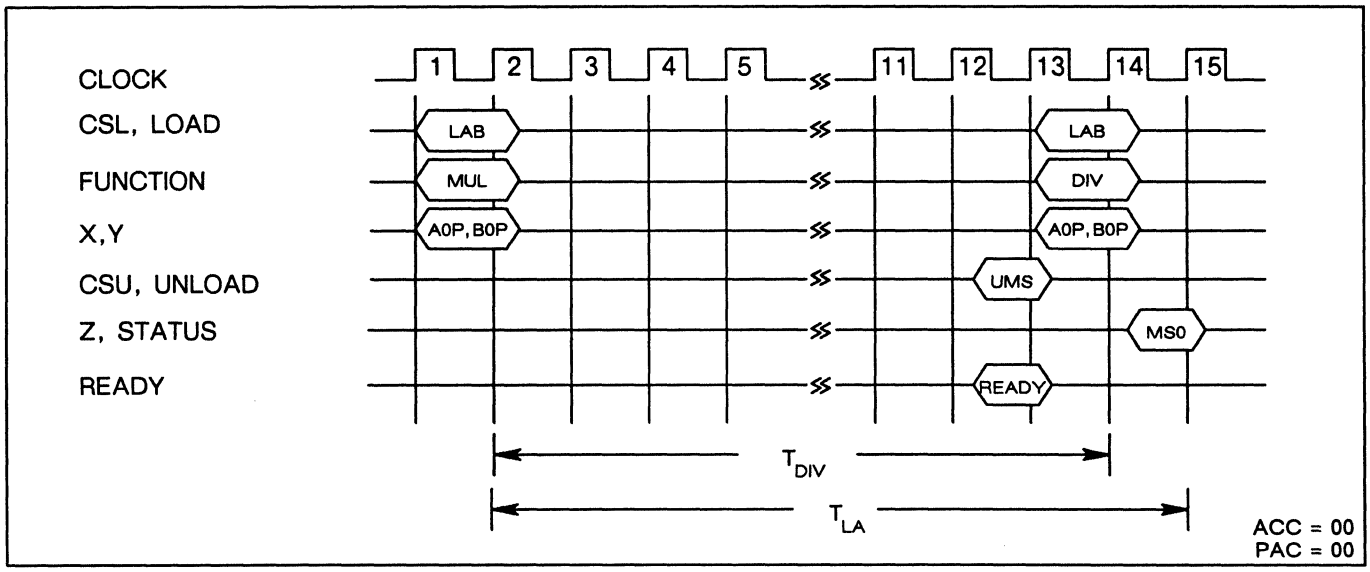


Figure 9. WTL 2264-50, -60 or -75 32-bit Divide Operation

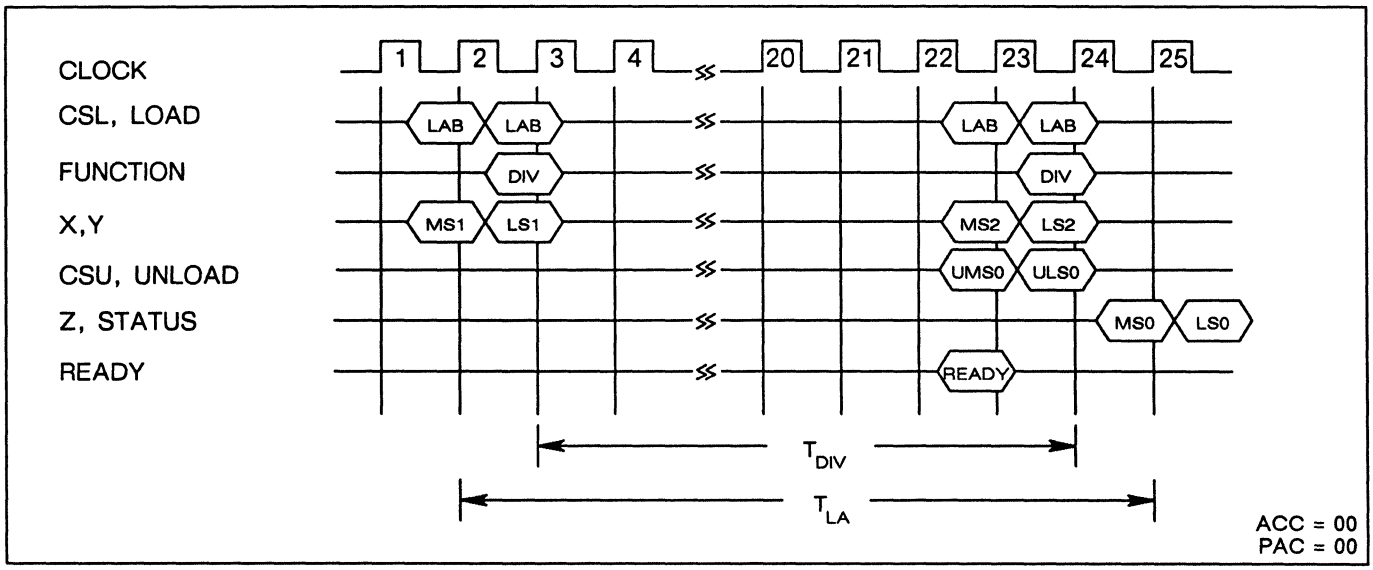


Figure 10. WTL 2264-50, -60 or -75 64-bit Divide Operation

Timing, continued

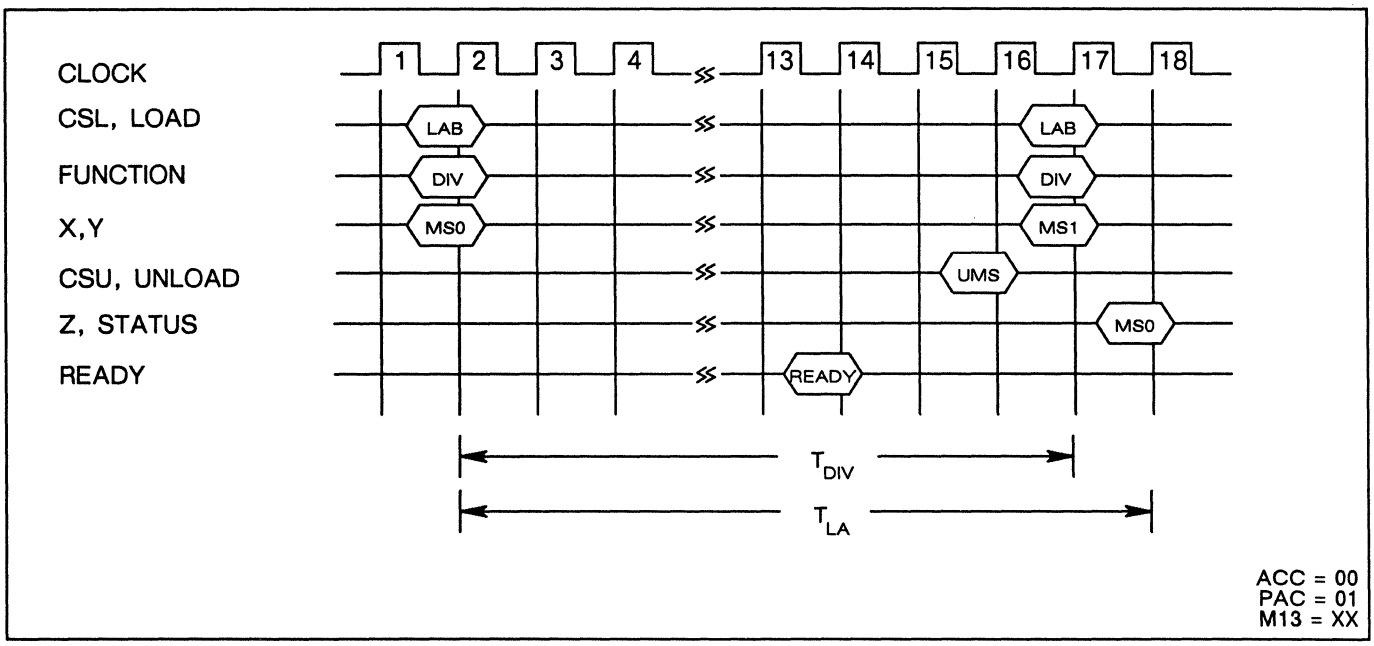


Figure 11. WTL 2264-80 or -100 32-bit Divide Operation

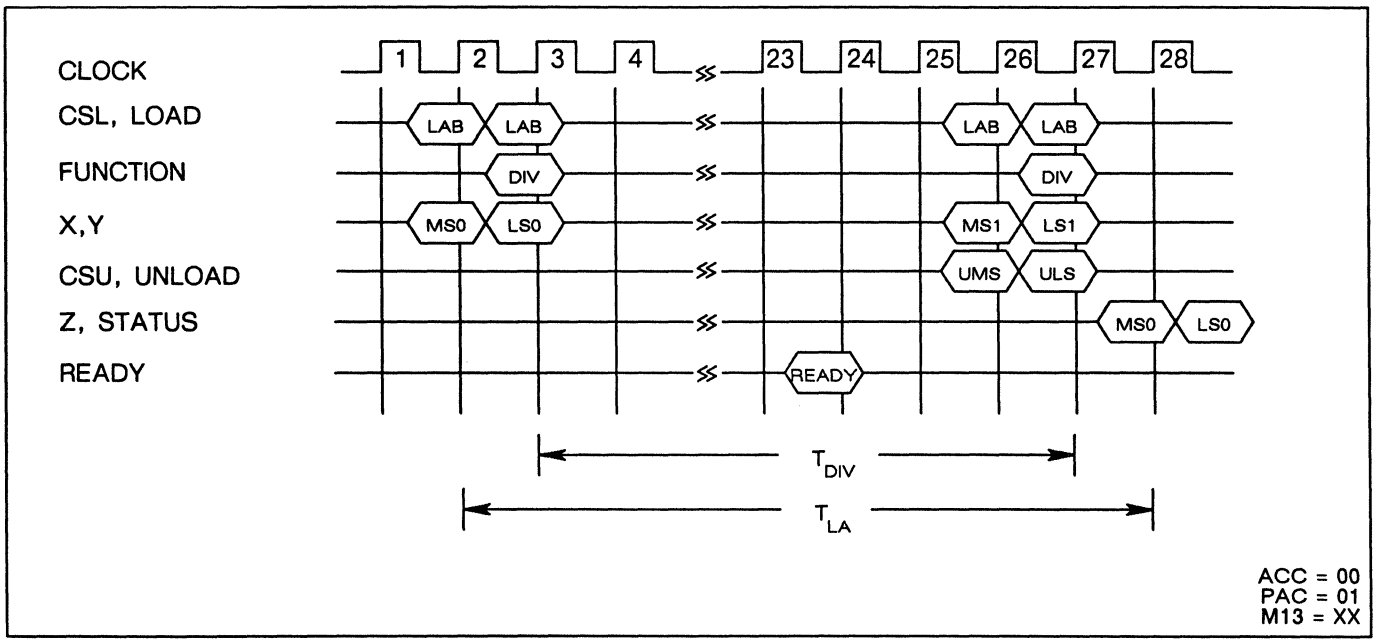


Figure 12. WTL 2264-80 or -100 64-bit Divide Operation

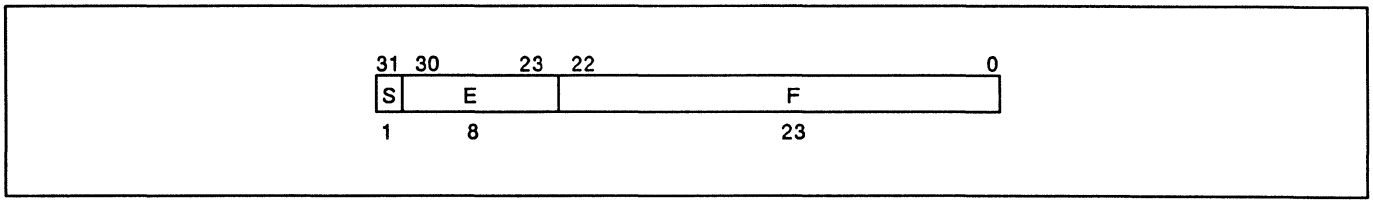
IEEE Compatibility, continued

IEEE Standard 754, Version 10.0 specifies floating point processor data formats, rounding modes and exception handling. The WTL 2264 and WTL 2265 conform to the specification. The discussion below reviews IEEE 754 implementation on the WTL 2264/2265. A separate note describes how denormalized numbers (DNRMs) are handled.

DATA FORMATS

The WTL 2264/2265 perform both 32-bit and 64-bit IEEE standard floating point operations. The 32-bit format has a 24-bit sign-magnitude fraction field and an 8-bit exponent, in the following format:

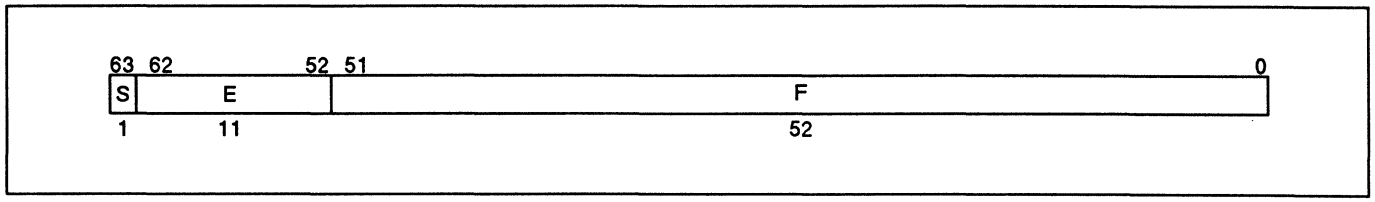
IEEE Standard Single Precision



Exponent values for normalized single precision numbers range from one to 254, with exponents of zero and 255 reserved for special operands. To calculate the value of a number in this format, the exponent is decremented by 127 (the "exponent bias" is +127), and the fraction has a one inserted before the binary point. (This is called the hidden bit.) The value of the number is then $(-1)^{S} \cdot 2^{e-127} \cdot (1.f)$

The 64-bit format has a 53-bit signed-magnitude fraction field and an 11-bit exponent, in the following format:

IEEE Standard Double Precision



Exponent values for normalized double precision numbers range from one to 2,046 with exponents of zero and 2,047 being reserved for special operands. To calculate the value of a number in this format, the exponent is decremented by 1,023 (the exponent bias is +1,023), and the fraction has a one inserted before the binary point. Thus the value of a double precision number is $(-1)^{S} \cdot 2^{e-1023} \cdot (1.f)$

Several number types are required to implement in the standard. These are normalized numbers, denormalized numbers, wrapped numbers, infinity and ZERO.

Normalized Numbers (NRM)

Most caculations are performed on normalized numbers. For single precision, normalized numbers have an exponent that ranges from 00000001 to 11111110 (one to 254) and a normalized fraction field (the leftmost or hidden bit is a one). In decimal notation, this allows one to represent a range of both positive and negative numbers from roughly 10^{38} to 10^{-38} with accuracy to seven decimal places. Double precision numbers have an exponent ranging from one to 2,046 and a normalized fraction field.

IEEE Compatibility, continued

Infinity (INF)

Infinity has an exponent of all ones and a fraction field equal to zero. Both positive and negative infinity are allowed.

ZERO

ZERO has an exponent of zero, a hidden bit equal to zero and a value of zero in the fraction field. Both +0 and −0 are supported.

Wrapped Numbers (WNRM)

A wrapped number is created by normalizing a DNRMs fraction field and subtracting from the exponent the number of shift positions required. (Normalizing is accomplished by left shifting until the hidden bit contains a one.) The value of the exponent is equal to (1 − (the number of shifts)) and is represented in two's complement.

Unrounded Normalized Number (UNRM)

A UNRM is the result of an operation that has magnitude less than the minimum representable normalized number. A UNRM has a normalized fraction field, a wrapped exponent and a hidden bit equal to one. The minimum UNRM is attained by multiplying two DNRM.MINs. UNRMs are turned into DNRM values using the ALU's unwrap function.

Denormalized Numbers (DNRM)

Denormalized numbers have a zero exponent and a denormalized (hidden bit equal to zero) non-zero fraction field.

Not A Number (NaN)

NaN is a special data format usually used as a flag for data flow control, for uninitialized variables or to signify an invalid operations such as $0 \bullet \infty$. The format for a NaN is an exponent of all ones and a non-zero fraction. The NaN produced by the WTL 2264/2265 has a sign bit of zero and fraction and exponent fields of all ones.

IEEE SINGLE PRECISION FORMATS SUPPORTED BY THE WTL 2264 AND WTL 2265

OPERAND	EXPONENT	FRACTION	HIDDEN BIT	VALUE
NaN	255	ANY	N/A	NONE
INFINITY	255	ALL 0's	1	$(-1)^s \infty$
NORM.MAX	254	ALL 1's	1	$(-1)^s \times 2^{127} \times (2)$
NORM	1 to 254	ANY	1	$(-1)^s \times 2^{e-127} \times (1.f)$
NORM.MIN	1	ALL 0's	1	$(-1)^s \times 2^{-126} \times (1)$
DNRM.MAX	0	ALL 1's	0	$(-1)^s \times (2^{-126} - 2^{-149})$
DNRM	0	ANY	0	$(-1)^s \times 2^{-126} \times (0.f)$
DNRM.MIN	0	000...01	0	$(-1)^s \times 2^{-126} \times 2^{-23}$
WNRM.MAX	0	ALL 1's	1	$(-1)^s \times 2^{-126}$
WNRM	0 to (−22)	ANY	1	$(-1)^s \times 2^{e-127} \times (1.f)$
WNRM.MIN	−22	ALL 0's	1	$(-1)^s \times 2^{-149}$
UNRM.MAX	0	ALL 1's	1	$(-1)^s \times 2^{-126}$
UNRM.MIN	−171	ALL 0's	1	$(-1)^s \times 2^{-298}$
ZERO	0	ALL 0's	0	$(-1)^s 0$

IEEE Compatibility, continued

The same formats are supported in double precision. The range of the double precision numbers and their values are obtained by substituting the double precision mantissa and exponent in the pattern shown above. A partial table of values is given below.

E	F	VALUE	NAME	MNEMONIC
2047	Not all zeros	None	Not a number	NaN
2047	All zeros	$(-1)^S *$ Infinity	Infinity	INF
1-2046	Any	$(-1)^S * 2^{E-1023} * (1.F)$	Normalized number	NOR
0	Not all zeros	$(-1)^S * 2^{-1022} * (0.F)$	Denormalized number	DNRM
0	Zero	$(-1)^S * 0$	Zero	ZERO

ROUNDING OPTIONS

The WTL 2264/2265 support all four rounding modes of the IEEE standard — round to nearest, round toward zero, round toward plus infinity and round toward minus infinity. Rounding may be biased or unbiased. Biased rounding introduces a small offset in the direction of the bias. Positive bias, negative bias or a bias toward zero are specified in the IEEE format. Unbiased rounding rounds the result to the nearest representable number. In the case of a number exactly halfway between two representable numbers, the number is rounded toward the closest even number, resulting in half the numbers rounding up and half rounding down, on average.

Round To Nearest

Rounds the result to the nearest representable value. If two numbers are equally near the result, the even number is chosen.

Round Toward Zero

Rounds the result to the value closest to but not greater than the magnitude of the result.

Round Toward Plus Infinity

Rounds the result to the value closest to but not less than the result.

Round Toward Minus Infinity

Rounds the result to the value closest to but no greater than the result.

EXCEPTION HANDLING

The WTL 2264/2265 generate the exceptions specified in the IEEE standard for floating point operations. The status word corresponding to an operation is propagated through the array and pipeline registers in synchrony with the operands and partial results. The status outputs are registered when the output data is clocked into the output register and is valid until the next rising edge of the clock. The status outputs also indicate the result of a COMPARE operation. Thus a COMPARE precludes the indication of other exceptions.

Inexact (NXT)

NXT is generated on the WTL 2264/2265 whenever there is a loss of accuracy. The chips compute results to higher precision than the number of mantissa bits that appear in the result. If any of the fraction bits less than the LSB was equal to one prior to rounding, the inexact bit will be high. NXT will be signaled in the WTL 2265 if there is a partial or complete loss of significance in a float-to-fixed operation.

Divide By Zero (DVZ)

The WTL 2265 will assert a DVZ exception when performing division on a normalized dividend and a zero divisor. The result is a properly signed infinity.

IEEE Compatability

Overflow (OVF)

OVF is generated when the result of a floating point operation overflows the largest representable normalized number. The result produced at the output is either infinity or the largest representable positive or negative number, depending upon the rounding mode as follows:

+MAX.NRM	if ((RM or RZ) and the result is positive)
-MAX.NRM	if ((RP or RZ) and the result is negative)
$+\infty$	if ((RN or RP) and the result is positive)
$-\infty$	if ((RN or RM) and the result is negative)

Overflow is also generated when converting floating-point-to-fixed-point and the result overflows the 32-bit format.

Underflow (UNF)

When the result of an operation after rounding is less than the minimum normalized number, UNF is asserted. A result of exactly zero does not underflow.

Invalid Operation (INV)

INV will be signaled in the WTL 2264/2265 if either input is a NaN (the status code will distinguish which) or if an invalid operation occurs. Operations invalid in the WTL 2264 are $(0 \bullet \infty)$, $(\infty \div \infty)$ or $(0 \div 0)$. Operations invalid in the WTL 2265 are subtraction of like infinities $(\infty - \infty)$ or addition of opposite infinities $(-\infty + \infty)$. For both the WTL 2264 and WTL 2265 the result of an INV is a NaN with fraction and exponent fields of all ones. The sign bit is zero.

Denormalized Input (DIN)

DIN is asserted whenever an operand is denormalized and the chip is in the IEEE mode. DIN applies only to the WTL 2264.

GRADUAL UNDERFLOW

The minimum normalized number has an exponent of one and a fraction field of zero. Zero has an exponent of zero and a fraction field of all zeros. This gives users the ability to deal with numbers between NORM.MIN and ZERO. These numbers are known as denormals. Their format is given in the number format section. The IEEE standard has specified gradual underflow to handle denormals. Many of the WTL 2264/2265's features are included to deal with denormals in a manner consistent with IEEE Standard 754, Version 10.0. Since denormals are very close to zero, many applications can substitute zero for a denormal without a signifcant loss of accuracy. For these applications, a "FAST" mode is included which substitutes zero for all denormalized inputs to the WTL 2264/2265. Zero is also inserted for all UNRM outputs in "FAST" mode.

For all arithmetic operations, the WTL 2265 handles denormalized inputs directly as it would handle any other number.

Unfortunately, a floating point multiplier must either operate exclusively on normalized numbers or suffer large cost and performance penalties in dealing directly with denormals. A normalized format that yields an equivalent to a given denormalized number is the wrapped format. The number format table shows the equivalence of wrapped and denormalized numbers. To translate a denormalized number to a wrapped number, the fraction is normalized (shifted up so that a one is in the hidden bit) and one is subtracted from the exponent for every position shifted. The WTL 2264 can multiply correctly either two wrapped numbers or a wrapped and a normalized number. To better understand the full procedure, consider the following case.

Assume one of the two input operands to the WTL 2264 is a denormalized number. Four cycles after the input, the denorm exception is flagged. The denormalized operand must then be sent to the WTL 2265 to be wrapped. Once wrapped, the operand can be sent back to the WTL 2264 for multiplication. The result of the multiplication will either be a normalized number or a UNRM.

If the result is a UNRM, status bit S_0 indicates either UNF (if all the truncated bits are equal to zero) or UNFNXT (if any of the truncated bits is equal to one).

No rounding will occur regardless of the rounding mode specified.

The underflowed number may then be sent to the WTL 2265 for "unwrapping". To unwrap a number, the fraction field is shifted right and the exponent incremented by one for each shift position. Status bit S_0 must be used to conditionally execute the UNWRAP INEXACT or UNWRAP EXACT instruction. The rounding must be performed in the ALU. The unwrapping may have three possible results:

IEEE Compatability, continued

RESULT	EXCEPTION	COMMENT
DNRM	UNF	When the denormalized result is exact. Note that his result is possible only if the UNWRAP EXACT instruction is possible (i. e., both the input and the result must be exact.)
DNRM	UNF–NXT	If the UNWRAP INEXACT instruction is executed *or* if the result of the UNWRAP EXACT instruction is inexact.
ZERO	NXT	The result is zero, but the unwrapping has resulted in the loss of precision.

Operations

The following tables delineate the results that are obtained for all combinations of input data formats and rounding options, for both the WTL 2264 and the WTL 2265 in IEEE as well as "FAST" mode. The format used in the tables is STATUS: (Status code)–Result.

TABLE 1: FLOATING POINT ADD/SUBTRACT ("FAST" MODE)

A/B	ZERO	DNRM	NRM	INF	NaN
NaN	INV:12–Nan	INV:12–NaN	INV:12–NaN	INV:12–NaN	INV:14–NaN
INF	OK*:1–INF	OK*:1–INF	OK*:1–INF	INV:15–NaN (2) OK*:1–INF (1)	INV:13–NaN
NRM	OK:2–NRM	OK:2,3–NRM	OVF:5–(4) OK:2,3–NRM UNF:6,7–ZERO OK:0–ZERO	OK*:1–INF	INV:13–NaN
DNRM	OK:0–ZERO (3)	OK:0–ZERO	OK:2,3–NRM	OK*:1–INF	INV:13–NaN
ZERO	OK:0–ZERO (3)	OK:0–ZERO (3)	OK:2–NRM	OK*:1–INF	INV:13–NaN

*If an operand is INF, OK will be signaled rather than OVF (see Note 1)

Operations, continued

Notes:

1. +INF+INF → +INF
   -INF-INF → -INF

2. +INF-INF → NaN (invalid operation)
   -INF+INF → NaN (invalid operation)

3. +ZERO+ZERO → +ZERO (RN, RZ, RP, RM)
   -ZERO-ZERO → -ZERO (RN, RZ, RP, RM)
   +ZERO-ZERO → +ZERO (RN, RZ, RP)
   +ZERO-ZERO → -ZERO (RM)
   -ZERO+ZERO → +ZERO (RN, RZ, RP)
   -ZERO+ZERO → -ZERO (RM)

4. OVF will produce INF or MAX.NRM, depending upon the rounding mode:
   +MAX.NRM IF [(RM, RZ) AND (RESULT IS +)]
   -MAX.NRM IF [(RP, RZ) AND (RESULT IS -)]
   +INF IF [(RN, RP) AND (RESULT IS +)]
   -INF IF [(RN, RM) AND (RESULT IS -)]

TABLE 2: FLOATING POINT MULTIPLICATION ("FAST" MODE)

A/B	ZERO	DNRM	NRM	INF	NaN
NaN	INF:12-NaN	INV:12-NaN	INV:12-NaN	INV:12-NaN	INV:14-NaN
INF	INV:15-NaN	INV:15-NaN	OK:1-INF	OK:1-INF	INV:13-NaN
NRM	OK:0-ZERO	OK:0-ZERO	OVF:5-(1) OK:2,3-NRM UNF:6,7-ZERO	OK:1-INF	INV:13-NaN
DNRM	OK:0-ZERO	OK:0-ZERO	OK:0-ZERO	INV:15-NaN	INV:13-NaN
ZERO	OK:0-ZERO	OK:0-ZERO	OK:0-ZERO	INV:15-NaN	INV:13-NaN

Notes:

1. OVF will produce INF or MAX.NRM, depending upon the rounding mode:
   +MAX.NRM IF [(RM, RZ) AND (RESULT IS +)]
   -MAX.NRM IF [(RP, RZ) AND (RESULT IS -)]
   +INF IF [(RN, RP) AND (RESULT IS +)]
   -INF IF [(RN, RM) AND (RESULT IS -)]

Operations, continued

TABLE 3: FLOATING POINT ADD/SUBTRACT (IEEE MODE)

A/B	ZERO	DNRM	NRM	INF	NaN
NaN	INF:12–NaN	INV:12–NaN	INV:12–NaN	INV:12–NaN	INV:14–NaN
INF	OK:1–INF	OK:1–INF	OK:1–INF	INV:15–NaN (2) OK:1–INF (1)	INV:13–NaN
NRM	OK:2–NRM	OVF:5–(4) OK:2,3–NRM UNF:6,7–UNRM	OVF:5–(4) OK:2,3–NRM UNF:6,7–UNRM OK:0–ZERO	OK:1–INF	INV:13–NaN
DNRM	UNF:6–UNRM	OK:0–ZERO (3) UNF:6–UNRM OK:2–NRM	OK:2,3–NRM UNF:6,7–UNRM OVF:5–(4)	OK:1–INF	INV:13–NaN
ZERO	OK:0–ZERO (3)	UNF:6–UNRM	OK:2–NRM	OK:1–INF	INV:13–NaN

Notes:

1. +INF+INF → +INF
   −INF−INF → −INF

2. +INF−INF → NaN
   −INF+INF → NaN

3. +ZERO+ZERO → +ZERO (RN, RZ, RP, RM)
   −ZERO−ZERO → −ZERO (RN, RZ, RP, RM)
   +ZERO−ZERO → +ZERO (RN, RZ, RP)
   +ZERO−ZERO → −ZERO (RM)
   −ZERO+ZERO → +ZERO (RN, RZ, RP)
   −ZERO+ZERO → −ZERO (RM)

4. OVF will produce INF or MAX.NRM, depending upon the rounding mode:
   +MAX.NRM IF [(RM, RZ) AND (RESULT IS +)]
   −MAX.NRM IF [(RP, RZ) AND (RESULT IS −)]
   +INF IF [(RN, RP) AND (RESULT IS +)]
   −INF IF [(RN, RM) AND (RESULT IS −)]

Operations, continued

TABLE 4: FLOATING POINT MULTIPLICATION (IEEE MODE)					
A/B	ZERO	DNRM	NRM	INF	NaN
NaN	INF:12–NaN	INV:12–NaN	INV:12–NaN	INV:12–NaN	INV:14–NaN
INF	INV:15–NaN	OK:1–INF	OK:1–INF	OK:1–INF	INV:13–NaN
NRM	OK:0–ZERO	DIN:9–(2)	OVF:5–(1) OK:2,3–NRM UNF:6,7–UNRM	OK:1–INF	INV:13–NaN
DNRM	OK:0–ZERO	DIN:10–(2)	DIN:8–(2)	OK:1–INF	INV:13–NaN
ZERO	OK:0–ZERO	OK:1–ZERO	OK:1–ZERO	INV:15–NaN	INV:13–NaN

Notes:

(1) OVF will produce INF or MAX.NRM, depending upon the rounding mode:

+MAX.NRM IF [(RM, RZ) AND (RESULT IS +)]
+MAX.NRM IF [(RP, RZ) AND (RESULT IS –)]
+INF IF [(RN, RP) AND (RESULT IS +)]
–INF IF [(RN, RM) AND (RESULT IS –)]

(2) Result is undefined.

TABLE 5: FLOATING POINT DIVISION ("FAST" MODE)					
A \ B	ZERO	DNRM	NRM	INF	NaN
NaN	INV:12–NaN	INV:12–NaN	INV:12–NaN	INV:12–NaN	INV:14–NaN
INF	OK:0–ZERO	OK:0–ZERO	OK:0–ZERO	INV:15–NaN	INV:13–NaN
NRM	OK:0–ZERO	OK:0–ZERO	OK:2,3–NRM OVF:5–(1) UNF:6,7–ZERO	OK:1–INF	INV:13–NaN
DNRM	INV:15–NaN	INV:15–NaN	DVZ:11–INF	OK:1–INF	INV:13–NaN
ZERO	INV:15–NaN	INV:15–NaN	DVZ:11–INF	OK:1–INF	INV:13–NaN

Operations, continued

Notes:
(1) Division takes the form B ÷ A

(2) OVF will produce INF or MAX.NRM, depending upon the rounding mode:

+MAX.NRM	IF	[(RM, RZ) AND (RESULT IS +)]
+MAX.NRM	IF	[(RP, RZ) AND (RESULT IS -)]
+INF	IF	[(RN, RP) AND (RESULT IS +)]
-INF	IF	[(RN, RM) AND (RESULT IS -)]

TABLE 6: FLOATING POINT DIVISION (IEEE MODE)

A \ B	ZERO	DNRM	NRM	INF	NaN
NaN	INV:12-NaN	INV:12-NaN	INV:12-NaN	INV:12-NaN	INV:14-NaN
INF	OK:0-ZERO	BDIN:9-(2)	OK:0-ZERO	INV:15-NaN	INV:13-NaN
NRM	OK:0-ZERO	BDIN:9-(2)	OK:2,3-NRM OVF:5-(1) UNF:6,7-UNRM	OK:1-INF	INV:13-NaN
DNRM	ADIN:8-(2)	ABDIN:10-(2)	ADIN:8-(2)	ADIN:8-(2)	INV:13-NaN
ZERO	INV:15-NaN	DVZ:11-INF	DVZ:11-INF	OK:1-INF	INV:13-NaN

Notes:

(1) OVF will produce INF or MAX.NRM, depending upon the rounding mode:

+MAX.NRM	IF	[(RM, RZ) AND (RESULT IS +)]
+MAX.NRM	IF	[(RP, RZ) AND (RESULT IS -)]
+INF	IF	[(RN, RP) AND (RESULT IS +)]
-INF	IF	[(RN, RM) AND (RESULT IS -)]

(2) Result is undefined.

Operations, continued

TABLE 7: FLOATING POINT COMPARE STATUS								
A/B	NaN	-INF	-NRM	-DNRM	ZERO	+DNRM	+NRM	+INF
NaN	U:15	U:15	U:15	U:15	U:15	U:15	U:15	U:15
+INF	U:15	G:2	G:2	G:2	G:2	G:2	G:2	E:0
+NRM	U:15	G:2	G:2	G:2	G:2	G:2	:0, 1, 2	L:1
+DNRM	U:15	G:2	G:2	G:2	G:2	:0, 1, 2	L:1	L:1
ZERO	U:15	G:2	G:2	G:2	E:0	L:1	L:1	L:1
-DNRM	U:15	G:2	G:2	:0, 1, 2	L:1	L:1	L:1	L:1
-NRM	U:15	G:2	:0, 1, 2	L:1	L:1	L:1	L:1	L:1
-INF	U:15	E:0	L:1	L:1	L:1	L:1	L:1	L:1

FORMAT: Condition: Status Code(s)

U:15 := unordered (status = 15)
E:0 := equal (status = 0)
L:1 := A < B (status = 1)
G:2 := A > B (status = 2)
:0, 1, 2 := may be A = B, A < B, or A > B, depending upon data values

TABLE 8: CONVERT SINGLE TO DOUBLE			
F32 → F64			
F32 OPERAND	F64 RESULT	STATUS	COMMENTS
7FFFFFFF or FFFFFFFF	7FFFFFFF FFFFFFFF	12	A operand is NaN
7F800000	7FF00000 00000000	1	+INF
7F7FFFFF	47EFFFFF E0000000	2	Input operand is +MAX.NRM
3F800000	3FF00000 00000000	2	+1
00800000	38100000 00000000	2	Input operand is +MIN.NRM
007FFFFF	380FFFFF C0000000	2	Input operand is +MAX.DNRM Result = 0 in "FAST" mode
00000001	36A00000 00000000	2	Input operand is +MIN.DNRM Result = 0 in "FAST" mode
00000000	00000000 00000000	0	+ZERO

Note: Sign bit is orthogonal; it is directly copied from the input operand to the output result (except for NaN which is clamped to zero).

Operations, continued

TABLE 9: CONVERT DOUBLE TO SINGLE			
F64 → F32 (Round to Nearest)			
F64 OPERAND	F32 RESULT	STATUS	COMMENTS
7FFFFFFF FFFFFFFF	7FFFFFFF	12	A operand is NaN
7FF00000 00000000	7F800000	1	+INF
7FEFFFFF FFFFFFFF	7F800000	5	+MAX.NRM OVERFLOWS
47EFFFFF F0000000	7F800000	5	+OVF RESULT
47EFFFFF E0000000	7F7FFFFF	2	+MAX.NRM RESULT
3FF00000 00000000	3F800000	2	+1
38100000 00000000	00800000	2	+MIN.NRM RESULT
380FFFFF FFFFFFFF	00800000	3	Result after rounding is +MIN.NRM
38000000 00000000	00000000 (UNRM) WTL 2265 00400000 (DNRM) WTL 1265	6 6	Result underflows
36FFFFFF FFFFFFFF	77FFFFFF (UNRM) WTL 2265 00000040 (DNRM) WTL 1265	7 7	Produces UNRM + NXT
36A00000 00000000	75000000 (UNRM) WTL 2265 00000001 (DNRM) WTL 1265	6 6	+MIN.DNRM RESULT
00000000 00000001	40000000 (UNRM) WTL 2265 00000000 (DNRM) WTL 1265	7 7	Input is DNRM
00000000 00000000	00000000	0	ZERO

Note: Sign bit is orthogonal; it is directly copied from the input operand to the output result (except for NaN which is clamped to zero).

Operations, continued

TABLE 10: DOUBLE FLOAT			
I32 → F64			
I32 OPERAND	F64 RESULT	STATUS	COMMENTS
7FFFFFFF	41DFFFFF FFC00000	2	Largest Positive Integer
00000001	3FF00000 00000000	2	+1
00000000	00000000 00000000	0	ZERO
FFFFFFFF	BFF00000 00000000	2	-1
80000000	C1E00000 00000000	2	Largest Negative Integer

TABLE 11: SINGLE FLOAT			
I32 → F32			
I32 OPERAND	F32 RESULT	STATUS	COMMENTS
7FFFFFFF	4F000000	3	Largest Positive Integer
7FFFFFC0	4F000000	3	INEXACT
7FFFFF80	4EFFFFFF	2	EXACT
00000001	3F800000	2	+1
00000000	00000000	0	ZERO
FFFFFFFF	BF800000	2	-1
80000080	CEFFFFFF	2	EXACT
80000040	CF000000	3	INEXACT
80000000	CF000000	2	Largest Negative Integer

Operations, continued

TABLE 12: DOUBLE FIX F64 → I32 (Round to Nearest)			
F64 OPERAND	I32 RESULT	STATUS	COMMENTS
7FFFFFFF FFFFFFFF	7FFFFFFF	12	Input is NaN
7FF00000 00000000	7FFFFFFF	5	+INF
7FEFFFFF FFFFFFFF	7FFFFFFF	5	Input is +MAX.NRM
41DFFFFF FFC00000	7FFFFFFF	2	Largest Positive Integer Result
3FF00000 00000000	00000001	2	+1
3FE80000 00000000	00000001	3	INEXACT
00100000 00000000	00000000	3	Input is MIN.NRM
00000000 00000001	00000000	3	Input is MIN.DNRM
00000000 00000000	00000000	0	+ZERO
80000000 00000000	00000000	0	-ZERO
8FF00000 00000000	00000000	3	Small Negative Number
BFF00000 00000000	FFFFFFFF	2	-1
C1E00000 00000000	80000000	2: WTL 2265 5: 1165/1265	Largest Negative Integer Result
FFF00000 00000000	80000000	5	-INF
FFFFFFFF FFFFFFFF	7FFFFFFF	12	-NaN

Operations, continued

TABLE 13: SINGLE FIX			
F32 → I32			
F32 OPERAND	I32 RESULT	STATUS	COMMENTS
7FFFFFFF	7FFFFFFF	12	Input is NaN
7F800000	7FFFFFFF	5	+INF
7F7FFFFF	7FFFFFFF	5	Input is +MAX.NRM
4F000000	7FFFFFFF	5	+OVF
4EFFFFFF	7FFFFF80	2	EXACT
3F800000	00000001	2	+1
3F400000	00000001	3	INEXACT
00800000	00000000	3	Input is +MIN.NRM
00000001	00000000	3	Input is +MIN.DNRM
00000000	00000000	0	+ZERO
80000000	00000000	0	-ZERO
8F800000	00000000	3	Small Negative Number
BF800000	FFFFFFFF	2	-1
CEFFFFFF	80000080	2	Large Negative Number
CF000001	80000000	5	-OVF
FF800000	80000000	5	-INF

TABLE 14: DOUBLE WRAP DENORMALIZED VALUE			
F64 → W64			
F64 OPERAND	W64 RESULT	STATUS	COMMENTS
00000000 00000001	7CD00000 00000000	6	Input is +MIN.DNRM
00080000 00000000	00000000 00000000	6	Always Exact
000FFFFF FFFFFFFF	000FFFFF FFFFFFFE	6	Input is +MAX.DNRM

TABLE 15: SINGLE WRAP DENORMALIZED VALUE			
F32 → W32			
F32 OPERAND	W32 RESULT	STATUS	COMMENTS
00000001	75000000	6	Input is -MIN.DNRM
00400000	00000000	6	Always Exact
007FFFFF	007FFFFE	6	Input is -MAX.DNRM

Operations, continued

TABLE 16: DOUBLE UNWRAP EXACT VALUE			
U64 → D64			
U64 OPERAND	D64 RESULT	STATUS	COMMENTS
000FFFFF FFFFFFFF	00100000 00000000	3	Result is NRM + NXT
00000000 00000000	00080000 00000000	6	UNF + EXACT
7FFFFFFF FFFFFFFF	00080000 00000000	7	UNF + NXT
40000000 00000000	00000000 00000000	7	UNF + NXT

TABLE 17: SINGLE UNWRAP EXACT VALUE			
U32 → D32			
U32 OPERAND	D32 RESULT	STATUS	COMMENTS
007FFFFF	00800000	3	Result is NRM
007FFFFE	007FFFFF	6	Result is DNRM
00000000	00400000	6	UNF + EXACT
7FFFFFFF	00400000	7	UNF + NXT
40000000	00000000	7	UNF + NXT

Note: For single and double unwrap functions, the sign bit of the output result is directly copied from the sign bit of the input operand.

I/O Characteristics

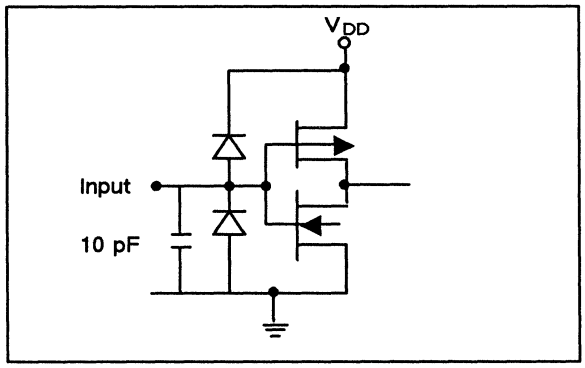


Figure 13. Input Equivalent Circuit

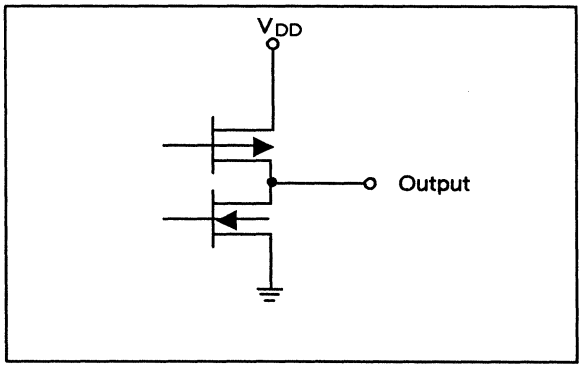


Figure 14. Output Equivalent Circuit

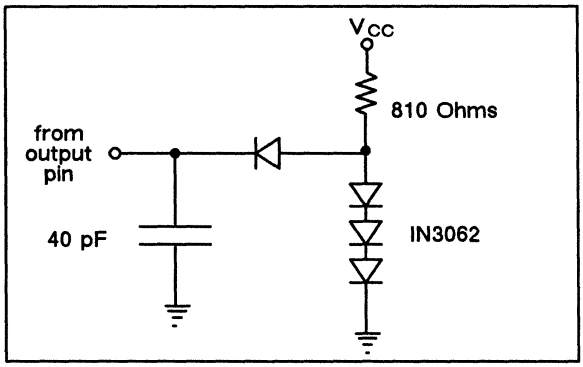


Figure 15. Normal Load Circuit for Delay Measurement

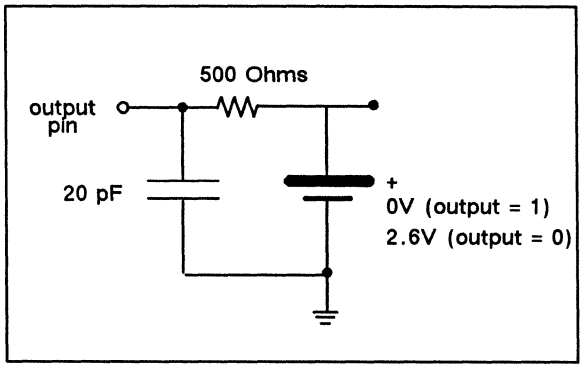


Figure 16. Tri-state Delay Load Circuit

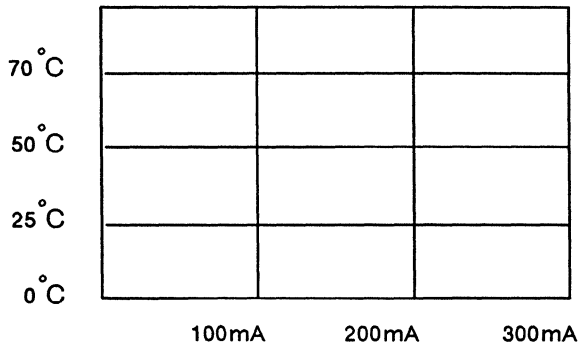


Figure 17. Temperature vs. Power Dissipation

I/O Characteristics, continued

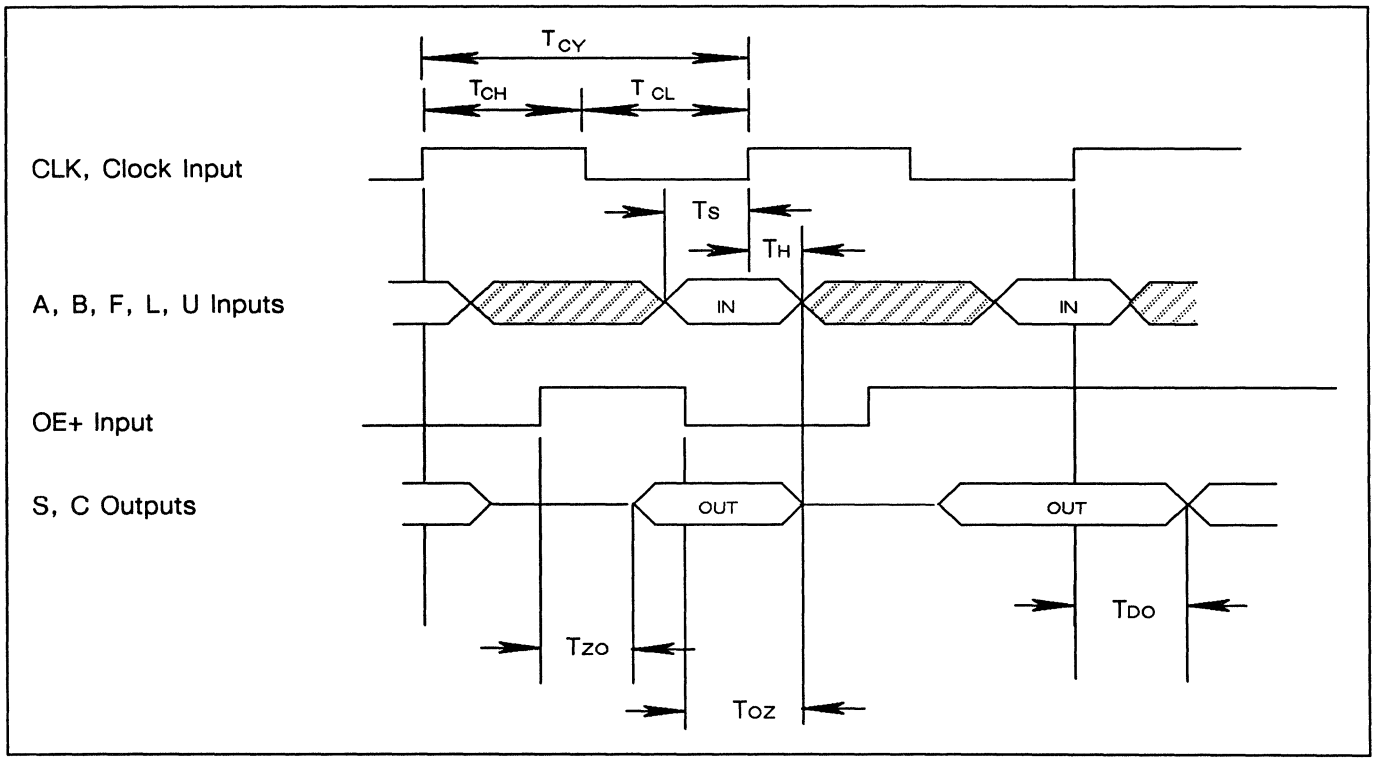


Figure 16. Input/Output Timing

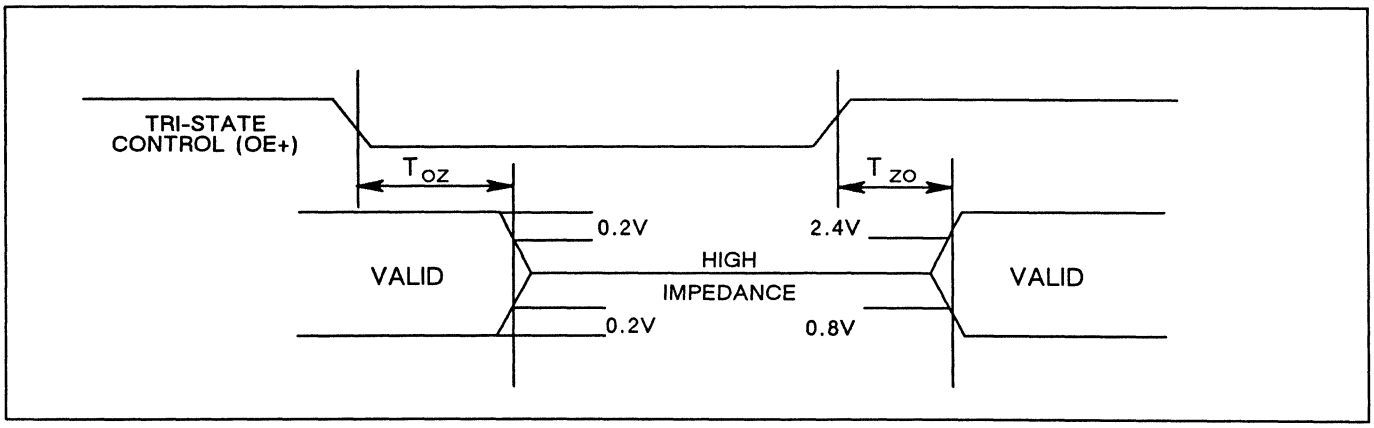
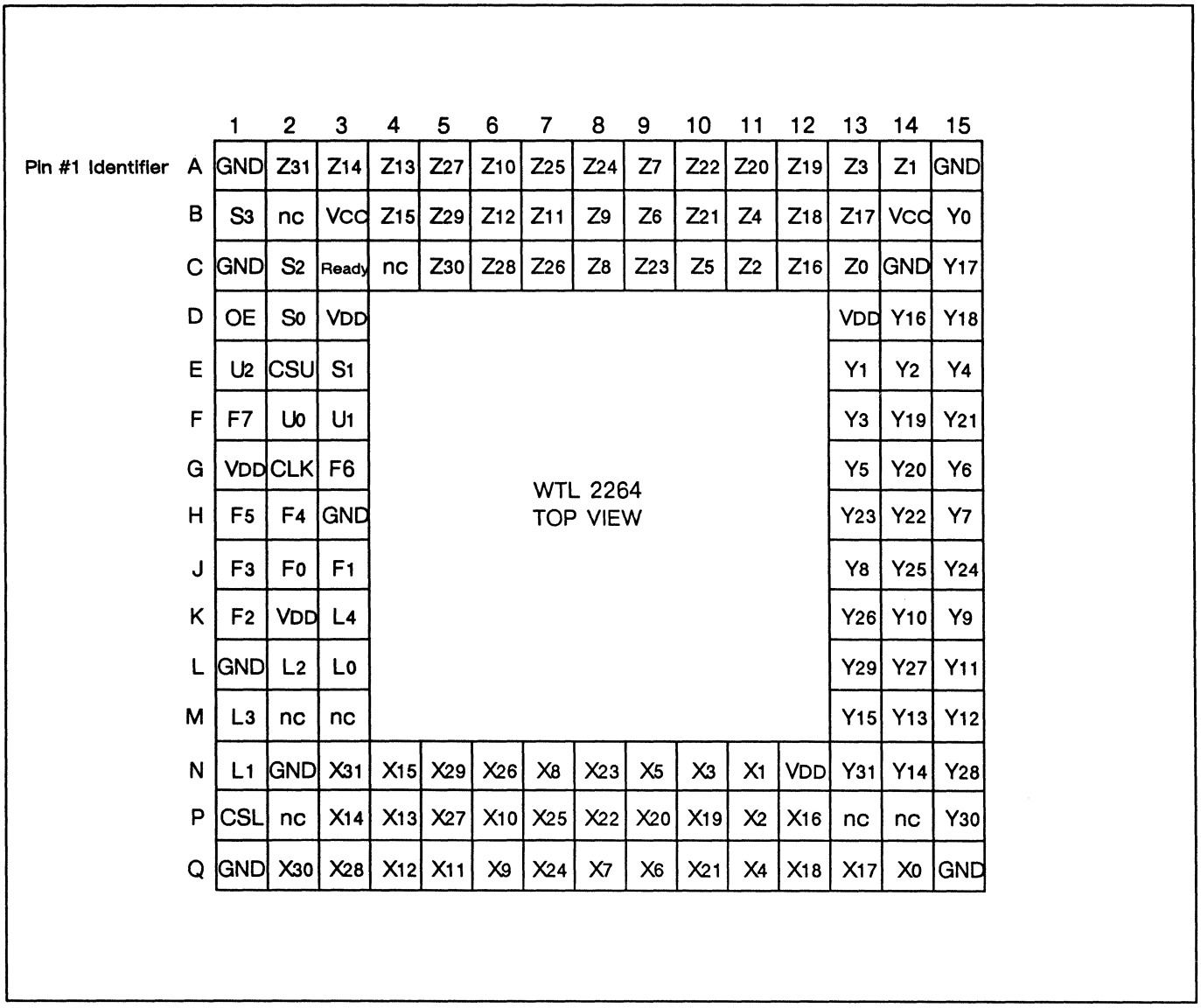


Figure 17. Tri-state Enable/Disable Timing Diagram

Pin Configuration



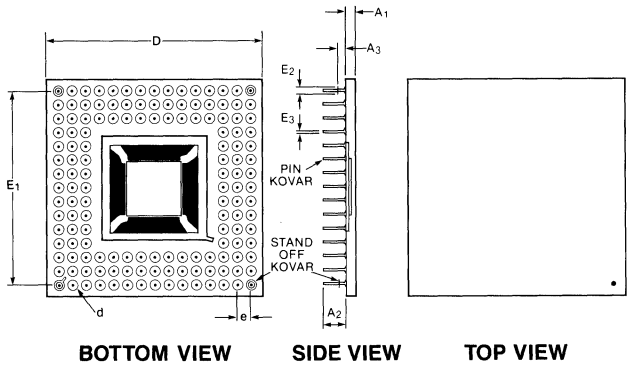
Pin #1 Identifier	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
A	GND	Z31	Z14	Z13	Z27	Z10	Z25	Z24	Z7	Z22	Z20	Z19	Z3	Z1	GND
B	S3	nc	VCC	Z15	Z29	Z12	Z11	Z9	Z6	Z21	Z4	Z18	Z17	VCC	Y0
C	GND	S2	Ready	nc	Z30	Z28	Z26	Z8	Z23	Z5	Z2	Z16	Z0	GND	Y17
D	OE	S0	VDD										VDD	Y16	Y18
E	U2	CSU	S1										Y1	Y2	Y4
F	F7	U0	U1										Y3	Y19	Y21
G	VDD	CLK	F6										Y5	Y20	Y6
H	F5	F4	GND										Y23	Y22	Y7
J	F3	F0	F1										Y8	Y25	Y24
K	F2	VDD	L4										Y26	Y10	Y9
L	GND	L2	L0										Y29	Y27	Y11
M	L3	nc	nc										Y15	Y13	Y12
N	L1	GND	X31	X15	X29	X26	X8	X23	X5	X3	X1	VDD	Y31	Y14	Y28
P	CSL	nc	X14	X13	X27	X10	X25	X22	X20	X19	X2	X16	nc	nc	Y30
Q	GND	X30	X28	X12	X11	X9	X24	X7	X6	X21	X4	X18	X17	X0	GND

WTL 2264
TOP VIEW

Pin Configuration, continued

Pin #1 Identifier	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
A	GND	Z31	Z14	Z13	Z27	Z10	Z25	Z24	Z7	Z22	Z20	Z19	Z3	Z1	GND
B	S3	nc	VCC	Z15	Z29	Z12	Z11	Z9	Z6	Z21	Z4	Z18	Z17	VCC	Y0
C	GND	S2	Ready	nc	Z30	Z28	Z26	Z8	Z23	Z5	Z2	Z16	Z0	GND	Y17
D	OE	S0	VDD										VDD	Y16	Y18
E	U2	CSU	S1										Y1	Y2	Y4
F	GND	U0	U1										Y3	Y19	Y21
G	VDD	CLK	F6										Y5	Y20	Y6
H	F5	F4	GND				WTL 2265 TOP VIEW						Y23	Y22	Y7
J	F3	F0	F1										Y8	Y25	Y24
K	F2	VDD	L4										Y26	Y10	Y9
L	L5	L2	L0										Y29	Y27	Y11
M	L3	nc	nc										Y15	Y13	Y12
N	L1	GND	X31	X15	X29	X26	X8	X23	X5	X3	X1	VDD	Y31	Y14	Y28
P	CSL	nc	X14	X13	X27	X10	X25	X22	X20	X19	X2	X16	nc	nc	Y30
Q	GND	X30	X28	X12	X11	X9	X24	X7	X6	X21	X4	X18	X17	X0	GND

Physical Dimensions



Symbol	DIMENSIONS	
	INCHES	MM
A1*	0.091 ± 0.010	1.31 ±0.25
A2	0.180 typ.	4.57
A3	0.050	1.27
D	1.575 sq. ± 0.016	40.0 ±.41
E1	1.400 sq. ± 0.012	35.56 ±.30
E2	0.050 dia. typ.	1.27
E3	0.018 ±0.002	.46 ± .05
d	0.070 dia. typ.	1,78
e	0.100 typ.	2.54

*A1 = package plus lid

Application Note

COMPATABILITY WITH WTL 1264 AND WTL 1265

There are two common WTL 1264/1265 configurations. In the first, one WTL 1264 is used per each WTL 1265. In the second, two WTL 1264 devices are used for each WTL 1265. We will discuss the 1:1 configuration. Contact WEITEK for a description of the 2:1 configuration.

Several programming models are used in this first configuration. As discussed in COMPATABILITY MODE, the most common programming model for the WTL 1264/1265 is maximum 64-bit throughput. If the compatability mode bit, M_{12}, is set to "0" the WTL 2264/2265 can operate in the WTL 1264/1265 programming mode. The effect of M_{12} is different for the WTL 2264 and WTL 2265.

If M_{12} is "0" in the WTL 2265, M_5 can be used to control the WTL 2265's dummy pipes in the same way it controls Pipe 3 in the WTL 1265. In the WTL 2264, M_{12} set to "0" causes PAC to be 01, ACC to be 01 and M_{13} to be "1". The following tables illustrate the WTL 2264/2265 mode configuration for several other WTL 1264/1265 timing models.

TABLE 18: WTL1264/WTL2264 COMPATABILITY CHART

	WTL 1264 MODE SETTINGS			COMPATIBLE WTL 2264-50 and -60 MODE SETTINGS				
Operation	M_{5-4}	M_{11-8}	M_{7-6}	M_{12}	$M_{5,4,1}$	M_{9-8}	M_{13}	M_{7-6}
	Pipeline Configuration	Pipe Advance	Accumulator		Pipeline Configuration	Pipe Advance	Pipe 2 Advance	Accumulator
64-bit max throughput	10	0100	01	0	111*	XX	X	XX
32-bit max throughput	11	0010	01	1	111	01	0	01
64-bit min latency	00	0000	01	1	100	01	X	01
32-bit min latency	00	0000	01	1	100	00	X	01

*Setting M_{12} to 0 causes $M_1 = 1$, $M_{9-8} = 01$, $M_{13} = 1$, $M_{7-6} = 01$.

Application Note, continued

TABLE 19: WTL 1265/WTLL 2265 COMPATABILITY CHART					
	WTL 1265 MODE SETTINGS		COMPATIBLE WTL 2265-50 and -60 MODE SETTINGS		
Operation	M_{7-4}	M_{11-8}	M_{12}	M_{7-4}	M_{9-8}
	Pipeline Configuration	Pipe Advance		Pipeline Configuration	Pipe Advance
64-bit max throughput	1111	0010	0	1111	01
32-bit max throughput	1111	0010	0	1111	01
64-bit min latency	0000	0000	1	1000	01
32-bit min latency	0000	0000	1	1000	00

Ordering Information

PACKAGE TYPE	TEMPERATURE RANGE	ORDER NUMBER
144-Pin Grid Array	Tc = 0 to 85° C	WTL 2264-100-GC/WTL 2265-100-GC
144-Pin Grid Array	Tc = 0 to 85° C	WTL 2264-080-GC/WTL 2265-080-GC
144-Pin Grid Array	Tc = 0 to 85° C	WTL 2264-075-GC/WTL 2265-075-GC
144-Pin Grid Array	Tc = 0 to 85° C	WTL 2264-060-GC/WTL 2265-060-GC
144-Pin Grid Array	Tc = 0 to 85° C	WTL 2264-050-GC/WTL 2265-050-GC

Revision Summary

1.	AC Electrical Characteristics	Revised	pages 4–5
2.	Signal Description	Revised	page 8
3.	Load Controls for WTL 2265	Revised	page 10
4.	Function Controls for WTL 2264	Revised	pages 14–15
5.	Compatability Mode Control	New	pages 16–17
6.	Pipe 2 Advance Control	New	page 17
7.	Pipeline Advance Control	Revised	page 17
8.	Figure 6. WTL 2265-50, -60 or -75 Accumulate Timing	New	page 22
9.	Figure 7. WTL 2265-80 or -100 Accumulate Timing	New	page 23
10.	Division	Revised	pages 24–26
11.	Invalid Operation	Revised	page 30
12.	Operations	Revised	pages 31–41
13.	Table 5. Floating Point Division ("FAST" Mode)	New	page 34
14.	Table 6. Floating Point Division (IEEE Mode)	New	page 35
15.	WTL 2265 Pin Configuration	Revised	page 44
16.	Application Note	New	page 46

WTL 2264/WTL 2265
FLOATING POINT MULTIPLIER/ DIVIDER AND ALU

PRELIMINARY DATA

July 1986

For additional information on WEITEK products, please fill out the form below and mail.

Name Title

Company Phone

Address

Comments

I am currently involved in a design with the following Weitek products _____ and wish to be added to your design data base to insure that I receive status updates.

APPLICATION:

☐ ENGINEERING WORKSTATIONS
☐ GRAPHICS
☐ PERSONAL COMPUTERS
☐ SCIENTIFIC COMPUTERS
☐ OTHER _____

Check the products on which you wish to receive data sheets:

☐ Have a sales person call

ATTACHED PROCESSORS
☐ XL-SERIES OVERVIEW
☐ XL-8200 OVERVIEW

COPROCESSORS
☐ 1167
☐ 1164/1165
☐ 3164/3364
☐ 3167

BUILDING BLOCKS
☐ 2264/2265
☐ 3132/3332
☐ 1232/1233
☐ 1066
☐ 2010
☐ 2245
☐ 2516
☐ 2517

WEITEK use: Rec'd Out TPT Source: DS

Status

WEITEK WTL 2264/WTL 2265 FLOATING POINT
Please Comment On The Quality Of This Data Sheet.

Even though we have tried to make this data sheet as complete as possible, it is conceivable that we have missed something that may be important to you. If you believe this is the case, please describe what the missing information is, and we will consider including it in the next printing of the data sheet.

Fold, Staple and Mail to Weitek Corp.

NO POSTAGE
NECESSARY
IF MAILED
IN THE
UNITED STATES

BUSINESS REPLY MAIL

FIRST CLASS PERMIT NO. 1374 SUNNYVALE, CA

POSTAGE WILL BE PAID BY ADDRESSEE

WEITEK Corporation
1060 E. Arques Ave.
Sunnyvale, CA 94086-BRM-9759

ATTN: Ed Masuda

WEITEK

WEITEK'S CUSTOMER COMMITMENT:

Weitek's mission is simple: to provide you with VLSI solutions to solve your compute-intensive problems. We translate that mission into the following corporate objectives:

1. To be first to market with performance breakthroughs, allowing you to develop and market systems at the edge of your art.

2. To understand your product, technology, and market needs, so that we can develop Weitek products and corporate plans that will help you succeed.

3. To price our products based on the fair value they represent to you, our customers.

4. To invest far in excess of the industry average in Research and Development, giving you the latest products through technological innovation.

5. To invest far in excess of the industry average in Selling, Marketing, and Technical Applications Support, in order to provide you with service and support unmatched in the industry.

6. To serve as a reliable, resourceful, and quality business partner to our customers.

These are our objectives. We're committed to making them happen. If you have comments or suggestions on how we can do more for you, please don't hesitate to contact us.

Art Collmeyer
President

Headquarters
Weitek Corporation
1060 E. Arques Avenue
Sunnyvale, CA 94086
TWX 910-339-9545
WEITEK SVL
FAX (408) 738-1185
TEL (408) 738-8400

Domestic Sales Offices
Weitek Corporation
1060 E. Arques Avenue
Sunnyvale, CA 94086
TWX 910-339-9545
WEITEK SVL
FAX (408) 738-1185
TEL (408) 738-8400

Corporate Place IV
111 South Bedford St.
Suite 200
Burlington, MA 01803
FAX (617) 229-4902
TEL (617) 229-8080

European Sales Headquarters
Greyhound House, 23/24 George St.
Richmond, Surrey, TW9 1JY
England
TELEX 928940 RICHBI G
FAX 011-441 940 6208
TEL 011-441 549 0164

Japanese Representative
4-8-1 Tsuchihashi
Miyamae-Ku
Kawasaki, Kanagawa-Pre
213 Japan
FAX 044-877-4268
TEL 044-852-1135