

T49-12-05

CY7C602A

CYPRESS
SEMICONDUCTOR

Floating-Point Unit

Features

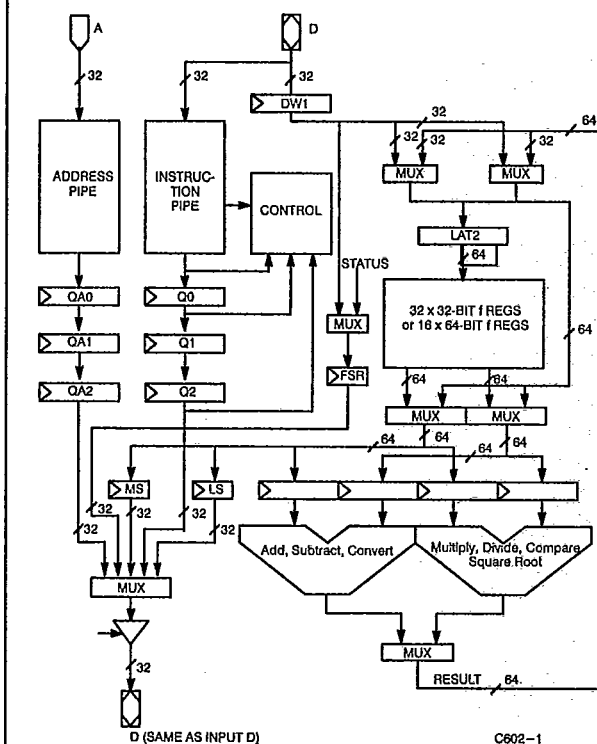
- Direct interface to CY7C601 integer unit
- Direct interface to CY7C157 Cache Storage Unit (CSU)
- Full compliance with ANSI/IEEE-754 standard for binary floating-point arithmetic
- Supports single and double precision floating-point operations
- 6.15 MFLOPs peak double-precision performance at 40 MHz
- SPARC-compatible interface allows concurrent execution of integer and floating-point instructions

- Hardware interlocks synchronize integer unit and floating-point unit operations
- 64-bit multiplier and divide/square root unit
- 64-bit ALU
- 16 64-bit registers or 32 32-bit registers in a three-port floating-point register file with an independent load/store port
- 144-pin PGA package
- Available in speeds of 25, 33, and 40 MHz

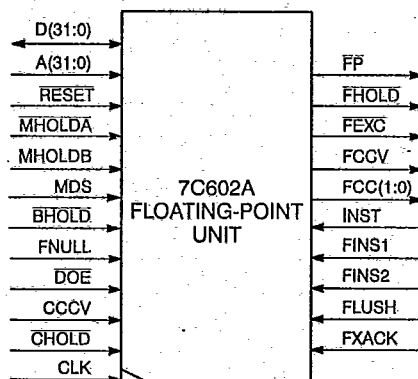
Description

The CY7C602A is a high-speed SPARC®-compatible floating-point unit for use with the CY7C601A integer unit. The CY7C602A floating-point unit allows floating-point instructions to execute concurrently with CY7C601A integer unit instructions. The CY7C602A interfaces directly to the CY7C601A integer unit without glue logic. The CY7C602A provides a peak 6.15 MFLOPs of double-precision performance at 40 MHz.

Logic Block Diagram



Pin Configuration



C602-2

Selection Guide

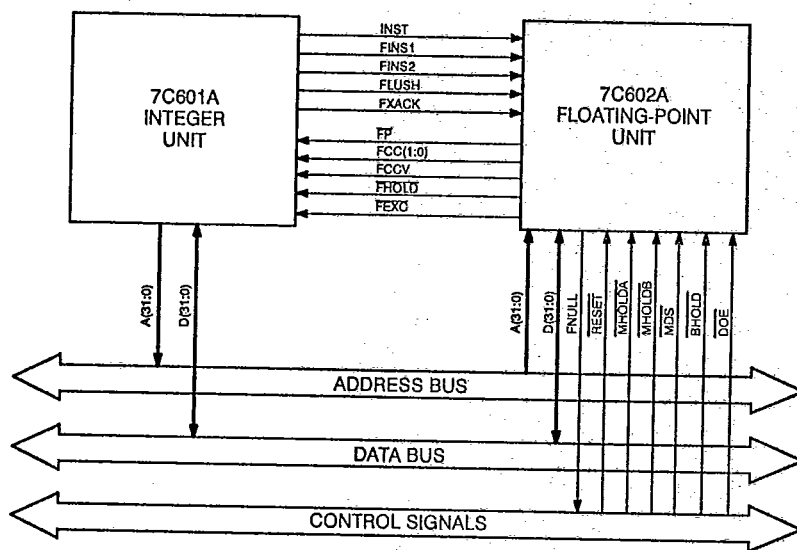
		7C602A-40	7C602A-33	7C602A-25
Maximum Supply Current (mA)	Commercial	450	400	350

SPARC is a registered trademark of SPARC International, Inc.



T-49-12-05

CY7C602A



C602-3

Figure 1. CY7C601A - CY7C602A Hardware Interface

Functional Description

The CY7C602A floating-point unit is a high-performance, single-chip implementation of the SPARC reference floating-point unit. The CY7C602A FPU directly interfaces with the CY7C601A integer unit, providing concurrent floating-point and integer instruction execution. The Cypress 7C600 chipset, comprised of the CY7C601A integer unit, CY7C602A floating-point unit, CY7C604A cache controller and memory management unit, and two CY7C157A CSUs, constitutes a high-performance CPU requiring no interface logic. The Cypress 7C600 chip-set is available in speeds up to 40 MHz, providing a sustained 29 MIPS of integer unit performance and over 6 MFLOPS of double-precision floating-point performance.

The CY7C602A supports single and double precision floating-point operation. Double precision floating-point is efficiently executed in the CY7C602A using a 64-bit internal datapath. The floating-point datapath circuitry contains a 64-bit multiplier, a 64-bit ALU, and a 64-bit divide/square-root unit. The CY7C602A provides thirty-two 32-bit floating-point registers, which can be concatenated for use as 64-bit registers. The CY7C602A complies with the ANSI/IEEE-754 floating-point standard.

The CY7C602A supports the execution of SPARC floating-point instructions. These instructions are separated into two groups: floating-point load/store and floating-point operate instructions (FPops). Floating-point load/store instructions are used to transfer data to and from the data registers (f registers). FP load/store instructions also allow the CY7C601A integer unit to read and write the floating-point status register (FSR) and to read the front entry of the floating-point queue. Floating-point operate instructions (FPops) include basic numeric operations (add, subtract, multiply, and divide), conversions between data types, register to register moves, and floating-point number comparison. FPops operate only on data in the floating-point registers. Floating-point branch instructions are executed by the IU on the basis of FP condition codes, and are not executed by the FPU.

The SPARC floating-point/integer unit interface provides concurrent execution of integer and floating-point instructions. The CY7C601A integer unit fetches all instructions for both itself and the CY7C602A FPU, providing all addressing and control signals. The CY7C602A floating-point unit latches all integer and floating-point instructions in parallel with the CY7C601A. When the CY7C601A decodes a floating-point instruction, it signals the CY7C602A with the FINS1 or FINS2 signal. This starts the execution of the floating-point instruction by the CY7C602A.

CY7C602A Registers

The CY7C602A has three types of user-accessible registers: the f registers, the FP queue, and the floating-point status register (FSR). The f registers are the CY7C602A data registers. The FSR is the CY7C602A status and operating mode register. The FP queue contains the CY7C602A instructions that have started execution and are awaiting completion. The following section describes these registers in detail.

f Registers

The CY7C602A provides 32 registers for floating-point operations, referred to as f registers. These registers are 32 bits in length, which can be concatenated to support 64-bit double words.

Integer and single precision data requires a single 32-bit f register. Double precision data requires 64 bits of storage and occupies an even-odd pair of adjacent f registers. Extended precision data requires 128 bits of storage and occupies a group of four consecutive f registers, always starting with register f0, f4, f8, f12, f20, f24, or f28.

The CY7C602A forces register addressing to match the data type specified by the floating-point instruction. This ensures data alignment in the f register file for double and extended precision data. Figure 2 illustrates how the CY7C602A uses the five



RISC

CYPRESS
SEMICONDUCTOR

T-49-12-05

CY7C602A

register address bits in a floating-point instruction for the different types of data. Single data word transfers (integer, single-precision floating-point) can be stored in any register. Consequently, all five bits of the register address specified in the floating-point instruction are valid. Double-precision data must reside in an even-odd pair of adjacent registers. By ignoring the LSB of the register address for a FPop requiring a register pair, the CY7C602A ensures data alignment. In a similar manner, the two LSBs of the register address are ignored in a SPARC FPU that supports extended precision data.

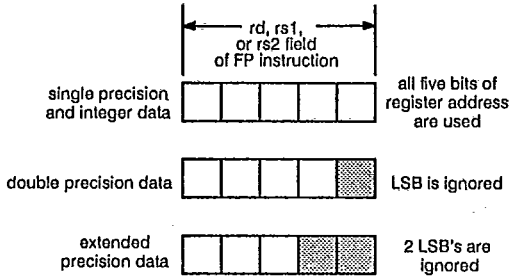


Figure 2. f Register Addressing

FP Queue

The CY7C602A maintains a floating-point queue of instructions that have started execution, but have yet to complete execution. The FP queue is used to accommodate the multiple clock nature of floating-point instructions. It also allows the CY7C602A to optimize execution through the use of data forwarding. Data forwarding allows FPop results to be used by a subsequent FPop before the results have been stored in its destination register. This saves one clock of execution time for each instruction that uses this feature.

The other purpose of the FP queue is to support the handling of FP exceptions. When the CY7C602A encounters an exception case, it enters pending exception mode and waits for the next FP instruction to be executed. When the CY7C601A decodes a FP instruction following the exception, it asserts the FINS1 or FINS2 signal. The CY7C602A then enters exception mode and asserts FEXC to signal a floating-point exception. When the CY7C602A enters the exception mode, floating-point execution halts until the FP queue is emptied. This allows the CY7C601A to store the floating-point instructions under execution when the exception case occurred. Emptying the FP queue frees the CY7C602A for use by the trap handler without losing the pre-exception state of the CY7C602A. After the trap handler finishes execution, the CY7C601A again fetches the FPop instructions previously stored in the FP queue, thus bringing the CY7C602A back to its previous state.

The FP queue contains the 32-bit address and 32-bit FPop instruction of up to three instructions under execution. Only FPop instructions are queued. The top entry of the FP queue is accessible by executing the store double floating-point queue (STDFQ) instruction. A load FP queue instruction does not exist, as the FP queue must be re-initialized by launching the queued instructions.

Floating-Point Status Register (FSR)

The following paragraphs describe the bit fields of the Floating-point status register (FSR). Figure 3 illustrates the bit assignments for the FSR. Refer to Table 1 (following page) for bit assignments for the FSR fields.

RD FSR(31:30). Rounding Direction: These two bits define the rounding direction used by the CY7C602A during an FP arithmetic operation.

RP FSR(29:28). Rounding Precision: These two bits define the rounding precision to which extended results are rounded. This is in accordance with the ANSI/IEEE STD-754-1985.

TEM FSR(27:23). Trap Enable Mask: These five bits enable traps caused by FPop. These bits are ANDed (1= enable, 0= disable) with the bits of the CEXC (current exception field) to determine which traps will force a floating-point exception to the CY7C601A. All trap enable fields correspond to the similarly named bit in the CEXC field (see below). The TEM field only affects which bits in the CEXC field will cause the FEXC signal to be asserted. ALL trap types, regardless of the state of the TEM field, are reported in the AEXC and CEXC fields.

NS FSR(22). Non-Standard Floating Point: This bit enables non-standard floating-point operations in the CY7C602A.

version FSR(19:17). The version number is used to identify the SPARC floating-point processor type. This field is set to 011 (3H) for the CY7C602A, and is read-only.

FTT FSR(16:14). Floating-point Trap Type: This field identifies the floating-point trap type of the current FP exception. This field can be read only.

QNE FSR(13). Queue Not Empty: This bit signals whether the FP queue is empty. (0= empty, 1= not empty)

FCC FSR(11:10). Floating-point Condition Codes: These two bits report the FP condition codes (see Table 1 below).

AEXC FSR(9:5). Accumulated EXceptions: This field reports the accumulated FP exceptions. All exception cases, masked or unmasked, are ORed with the contents of the AEXC and accumulated as status. All accumulated fields have the same definition as the corresponding field for CEXC (see below). This field can be read and written, and must be cleared by software (see Table 1).

CEXC FSR(4:0). Current EXceptions: This field reports the current FP exceptions. This field is automatically cleared upon the execution of the next floating-point instruction. CEXC status is not lost upon assertion of a floating-point exception, since instructions following a valid exception are not executed by the CY7C602A. The following defines the five CEXC bits:

nvc = 1 indicates invalid operation exception. This is defined as an operation using an improper operand value. An example of this is 0/0, ∞ , or $-\infty$.

ofc = 1 indicates overflow exception. The rounded result would be larger in magnitude than the largest normalized number in the specified format.

ufc = 1 indicates underflow exception. The rounded result is inexact, and would be smaller in magnitude than the smallest normalized number in the indicated format.

dzc = 1 indicates division-by-zero, X/0, where X is subnormal or normalized. Note that 0/0 does not set the dzc bit.

nxc = 1 indicates inexact exception. The rounded result differs from the infinitely precise correct result.

R FSR21, 20, and 12. Reserved - always set to 0.

CYPRESS
SEMICONDUCTOR

T-49-12-05

CY7C602A

RD	RP	TEM	NS	R	version	FTT	QNE	R	FCC	AEXC	CEXC	
31 30	29 28	27	23 22	21 20	19	17 16	14	13	12	11 10 9	5 4	0

TEM				
nvm	ofm	ufm	dzm	nxm

AEXC				
nva	ofa	ufa	dza	nxa

CEXC				
nvc	ofc	ufc	dzc	nxc

Figure 3. Floating-Point Status Register

Table 1. Floating-Point Status Register Summary

Field	Values	FSR bits	Description	Loadable by LDFSR
RD	0 - Round to nearest (tie-even) 1 - Round to 0 2 - Round to $+\infty$ 3 - Round to $-\infty$	31:30	Rounding Direction	yes
RP	0 - Extended precision 1 - Single precision 2 - Double precision 3 - Reserved	29:28	Extended Rounding Precision	yes
TEM	0 - Disable trap 1 - Enable trap NVM OFM UFM DZM NXM	27:23	Trap Enable Mask invalid operation trap mask overflow trap mask underflow trap mask divide by zero trap mask inexact trap mask	yes
NS	0 - Disable 1 - Enable	22	Non-standard Floating-point	yes
version	0 - 7	19:17	FPU version number	no
FTT	0 - None 1 - IEEE Exception 2 - Unfinished FPop 3 - Unimplemented FPop 4 - Sequence Error 5 - 7 Reserved	16:14	Floating-point trap type	no
QNE	0 - queue empty	13	Queue Not Empty	no
FCC	0 - = 1 - < 2 - > 3 - Unordered	11:10	Floating-point Condition Codes	yes
AEXC	NVA OFA UFA DXA NXA	9:5	Accrued Exception Bits accrued invalid exception accrued overflow exception accrued underflow exception accrued divide by zero exception accrued inexact exception	yes
CEXC	NVC OFC UFC DZC NXC	4:0	Current Exception Bits current invalid exception current overflow exception current underflow exception current divide by zero exception current inexact exception	yes
r	Always set to 0	21, 20, 12	reserved bits	no



RISC



CY7C602A Pin Definitions

Integer Unit Interface Signals:

FP active-low output. Floating-point Present: This signal indicates to the CY7C601A that a FPU is present in the system. In the absence of a FPU, this signal is pulled up to VCC by a resistor. This is a static signal; it always asserts a low output. The CY7C601A generates a floating-point disable trap if FP is not asserted during the execution of a floating-point instruction.

FCC(1:0) output. Floating-point Condition Codes: The FCC(1:0) bits indicate the current condition code of the FPU, and are valid only if FCCV is asserted. FBfcc instructions use the value of these bits during the execute cycle if they are valid. If the FCC bits are not valid, then FCCV is released, which halts the CY7C601A until the FCC bits become valid.

FCC1	FCC0	Condition
0	0	equal
0	1	Op1 < Op2
1	0	Op1 > Op2
1	1	Unordered

Table 2. FCC(1:0) Condition Codes

FCCV output. Floating-point Condition Codes Valid: The CY7C602A asserts the FCCV signal when the FCC represent a valid condition. The FCCV signal is deasserted when a pending floating-point compare instruction exists in the floating-point queue. FCCV is reasserted when the compare instruction is completed and FCC bits are valid.

FHOLD output. Floating-point HOLD: The FHOLD signal is asserted by the CY7C602A if it cannot continue execution due to a resource or operand dependency. The CY7C602A checks for all dependencies in the decode stage, and if necessary, asserts FHOLD in the next cycle. The FHOLD signal is used by the CY7C601A to freeze its pipeline in the same cycle. The CY7C602A must eventually deassert FHOLD to release the CY7C601A pipeline.

FEXC output. Floating-point EXception: The FEXC is asserted if a floating-point exception has occurred. It remains asserted until the CY7C601A acknowledges that it has taken a trap by asserting FXACK. Floating-point exceptions are taken only during the execution of a floating-point instruction. The CY7C602A releases FEXC when it receives FXACK.

FXACK input. Floating-point eXception ACKnowledge: The FXACK signal is asserted by the CY7C601A to acknowledge to the CY7C602A that the current FP trap is taken.

INST input. INSTRUCTION fetch: The INST signal is asserted by the CY7C601A whenever a new instruction is being fetched. It is used by the CY7C602A to latch the instruction on the D(31:0) bus into the FPU instruction buffer. The CY7C602A has two instruction buffers (D1 and D2) to save the last two fetched instructions. When INST is asserted, the new instruction enters the D1 buffer and the old instruction in D1 enters the D2 buffer.

FINS1 input. Floating-point INSTRUCTION in buffer 1: The FINS1 signal is asserted by the CY7C601A during the decode stage of a FPU instruction if the instruction is stored in the D1 buffer of the CY7C602A. The CY7C602A uses this signal to launch the instruction in the D1 buffer into its execute stage instruction register.

T-49-12-05

CY7C602A

FINS2 input. Floating-point INSTRUCTION in buffer 2: The FINS2 signal is asserted by the CY7C601A during the decode stage of a FPU instruction if the instruction is stored in the D2 buffer of the CY7C602A. The CY7C602A uses this signal to launch the instruction in the D2 buffer into its execute stage instruction register.

FLUSH input. Floating-point instruction FLUSH: The FLUSH signal is asserted by the CY7C601A to signal to the CY7C602A to flush the instructions in its instruction registers. This may happen when a trap is taken by the CY7C601A. The CY7C601A will restart the flushed instructions after returning from the trap. FLUSH has no effect on instructions in the floating-point queue. In addition to freezing the FPU pipeline, the CY7C602A uses FLUSH to shut off D bus drivers during store. To ensure correct operation of the CY7C602A, FLUSH must not change state more than once during a clock cycle.

Coprorocessor Interface Signals:

CHOLD input. Coprocessor HOLD: The CHOLD signal is asserted by the coprocessor if it cannot continue execution. The coprocessor must check all dependencies in the decode stage of the instruction and assert the CHOLD signal, if necessary, in the next cycle. The coprocessor must eventually deassert this signal to unfreeze the CY7C601A and CY7C602A pipelines. The CHOLD signal is latched with a transparent latch in the CY7C602A before it is used.

CCCV input. Coprocessor Condition Codes Valid: The coprocessor asserts the CCCV signal when the CCC(1:0) represent a valid condition. The CCCV signal is deasserted when a pending floating-point compare instruction exists in the coprocessor queue. CCCV is reasserted when the compare instruction is completed and CCC bits are valid. The CY7C602A will enter a wait state if CCCV is deasserted. The CCCV signal is latched with a transparent latch in the CY7C602A before it is used.

System/Memory Interface Signals:

A(31:0) input. Address bus (31:0): The address bus for the CY7C602A is an input-only bus. The CY7C601A supplies all addresses for instruction and data fetches for the CY7C602A. The CY7C602A captures addresses of floating-point instructions from the A(31:0) bus into the DDA register. When INST is asserted by the CY7C601A, the contents of the DDA is transferred to the DA1 register.

D(31:0) input/output. Data bus (31:0): The D(31:0) bus is driven by the FPU only during the execution of floating-point store instructions. The store data is sent out unlatched and must be latched externally before it is used. Once latched, store data is valid during the second data cycle of a store single access and on the second and third data cycle of a store double access. The data alignment for load and store instructions is done inside the FPU. A double word is aligned on an eight-byte boundary. A single word is aligned on a four-byte boundary.

DOE input. Data Output Enable: The DOE signal is connected directly to the data output drivers and must be asserted during normal operation. deassertion of this signal tri-states all output drivers on the data bus. This signal should be deasserted only when the bus is granted to another bus master, i.e., when either BHOLD, MHOLDA, or MHOLDB is asserted.

MHOLDA, MHOLDB input. Memory HOLD: Asserting MHOLDA or MHOLDB freezes the CY7C602A pipeline. Either MHOLDA or MHOLDB is used to freeze the FPU (and the



T-49-12-05

CY7C602A

IU) pipelines during a cache miss (for systems with cache) or when slow memory is accessed.

BHOLD input. Bus HOLD: This signal is asserted by the system's I/O controller when an external bus master requests the data bus. Assertion of this signal will freeze the FPU pipeline. External logic should guarantee that after deassertion of BHOLD, the state of all inputs to the chip is the same as before BHOLD was asserted.

MDS input. Memory Data Strobe: The MDS signal is used to load data into the FPU when the internal FPU pipeline is frozen by assertion of MHOLDA, MHOLDB, or BHOLD.

FNULL output. Fpu NULLify cycle: This signal signals to the memory system when the CY7C602A is holding the instruction pipeline of the system. This hold would occur when FHOLD or

FCCV is asserted. This signal is used by the memory system in the same fashion as the integer unit's INULL signal. The system needs this signal because the IU's INULL does not take into account holds requested by the FPU.

RESET input. RESET: Asserting the RESET signal resets the pipeline and sets the writable fields of the floating-point status register (FSR) to zero. The RESET signal must remain asserted for a minimum of eight cycles. After a reset, the IU will start fetching from address 0.

CLK input. CLOCK: The CLK signal is used for clocking the FPU's pipeline registers. It is high during the first half of the processor cycle and low during the second half. The rising edge of CLK defines the beginning of each pipeline stage in the FPU.

Document #: 38-R-10004-A



RISC