

# DSP5<sup>®</sup>

## High Performance 16Bit Fixed Point DSP Core

### General Description

Infinior Microsystems' DSP5<sup>®</sup> is a high performance 16-Bit fixed-point digital signal processor core. DSP5 is based on an advanced modified Harvard architecture that has one program memory bus and three data memory buses. This processor core provides an arithmetic logic unit (ALU) that has a high degree of parallelism, application-specific hardware logic, on-chip memory, and additional on-chip peripherals.

DSP5 also provides a highly specialized instruction set, which is the basis of the operational flexibility and speed of Digital Signal Processing.

DSP5 has been designed in synthesizable HDL (Hardware Description Language) and can be configured with variety of on-chip memories and peripherals.

DSP5 is proven at 0.35um static CMOS integrated-circuit technology. It is confirmed to execute more than 70 MIPS (Million Instructions Per Second). Core gate counts is up to 80K, which is not included memory modules.

DSP5 is fully synchronous and static design with single internal clock and operates in 6 stages pipelines (Pre-Fetch, Fetch, Decode, Access, Read, Execution).

DSP5 is targeted for high performance digital signal processing ASSP (Application Specific Standard Product) and can be used as an ASIC library element in a variety of DSP (Digital Signal Processor) applications and SOC (System On Chip) design.

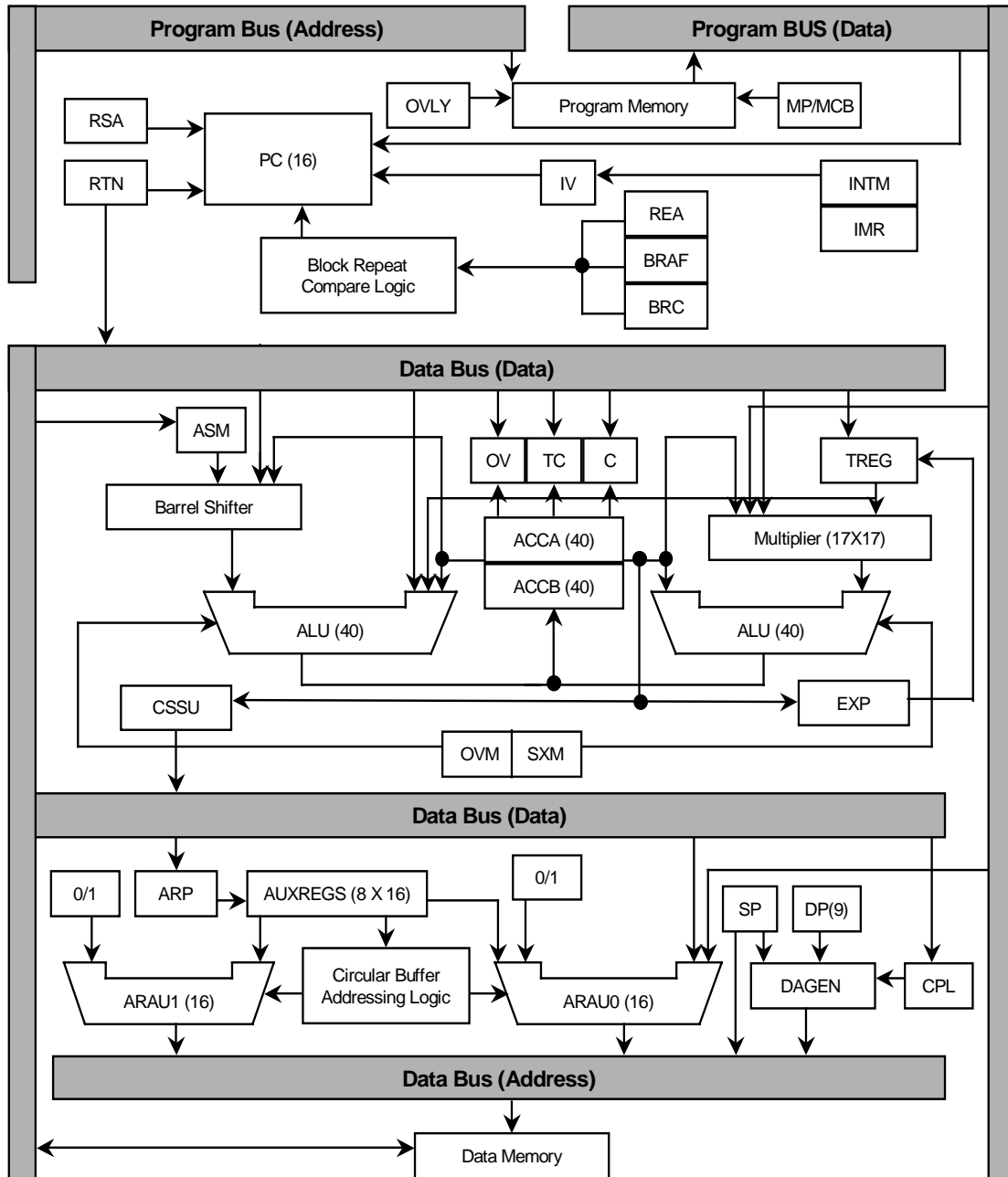
## Key Features

- 16 Bit Fixed Point Digital Signal Processor
- 6 Stage Pipeline (Prefetch, Fetch, Decoder, Access, Read, Execute)
- Advanced Multi-bus Harvard architecture with Three 16 Bit Data Buses and One Program Bus
- 64K words Program, Data, I/O Space
- 10K word internal program/data Two-Port RAM
- 40 Bit Arithmetic Logic Unit including One Barrel Shifter and Two independent 40-Bit Accumulators
- 17 x 17 Bit Parallel Multiplier Coupled to a 40 Bit Dedicated Adder for Non-Pipelined Single-Cycle Multiply/Accumulate (MAC) Operation
- Compare, Select, and Store Unit for the Add/Compare Selection of the Viterbi Operator
- Exponent Encoder to Compute an Exponent Value of a 40 Bit Accumulator Value in a Single Cycle
- Two Address Generators with Eight Auxiliary Registers and Two Auxiliary Register Arithmetic Units
- Instruction with 32 Bit Long Word Operand
- Single-Instruction Repeat and Block-Repeat Operations for Program Code
- Block-Memory-Move Instructions for Better Program and Data Management
- Conditional Store Instructions
- Arithmetic Instruction with Parallel Store and Parallel Load
- Buffered Serial Port, Time-Division Multiplexed Serial Port
- 8 Bit Parallel Host Port Interface
- Interrupt Controller, 16 Bit Timer, Wait State Generator
- Built-In on-chip Self Test programs
- Fully Synchronous and Static Design with 80k Core gates
- Supports Test Bench and Full Simulation and Synthesis Scripts

## Applications

- Multi-standard Audio Codec (MP3, WMA, AC3 etc.)
- VoIP Voice processor (G.729, G.723.1 etc.)
- Modem data pump (ITU-T V.34, G.DMT, G.VDSL)
- Speech recognition and synthesis

# Block Diagram



## Internal Blocks

The DSP5 utilizes a modified Harvard architecture for speed and flexibility. In a strict Harvard architecture, program and data memory lie in two separate spaces, permitting a full overlap of instruction fetch and execution. The DSP5's modification of the Harvard architecture allows transfers between program and data spaces, thereby increasing the flexibility of the device. This modification permits coefficients stored in program memory to be read into the RAM, eliminating the need for a separate coefficient ROM. It also makes available immediate instructions and subroutines based on computed values.

Two reads and one write operation can be performed in a single cycle. Instructions with parallel store and application-specific instructions can fully utilize this architecture. In addition, data can be transferred between data and program spaces. Such parallelism supports a powerful set of arithmetic, logic, and Bit-manipulation operations that can all be performed in a single machine cycle.

### (1) ALU

2's-complement arithmetic and boolean operations using a 40 Bit ALU and two 40 Bit Accumulators.

Two 16 Bit operations simultaneously using two 16 Bit ALUs.

### (2) Barrel Shifter

The 40 Bit barrel shifter produces a left shift of 0 to 31 Bits and a right shift of 0 to 16 Bits.

### (3) Multiplier/Adder

The multiplier/adder performs 17x17-Bit 2's-complement multiplication with a 40 Bit accumulation in a single instruction cycle.

The multiplier and ALU together execute multiply/accumulate (MAC) computations and ALU operations in parallel in a single instruction cycle.

### (4) Compare, select and store unit (CSSU)

The CSSU performs maximum comparisons between the ACC's high and low word, and selects the larger word in the accumulator to be stored in data memory.

**(5) Addressing Unit (AU)**

- Immediate Addressing ( ex : RPTK #8h )
- Absolute Addressing ( ex : LD \*(buffer), a )
- Accumulator Addressing ( ex : READA \*ar5 )
- Direct Addressing ( ex : ADD sample, b )
- Indirect Addressing ( ex : ADD \*ar2+0B, a )
- Memory-Mapped Register Addressing ( ex : STM #1000h, ar2 )
- Stack Addressing ( ex : PSHD \*ar2 )

**(6) Program Counter (PC)**

Branches, Calls, Returns, Conditional operations  
Repeats of an instruction or a block of instructions

**(7) Decoder**

Holds the pre-fetch instruction word from ROM for decoding and data address selection  
Containing status registers and Repeat counter

**(8) MMU (Internal and External Memory Management Unit)**

Memory management unit is for interfacing between core and memory  
With making only one RAM active, MMU decreases the power consumption

**(9) FSM**

Controls the sequence of execution of the current instruction

**(10) PLA**

The current instruction is taken from either the instruction buffer or from the instruction delay buffer. Provides a control of the current instruction word and state of the micro controller. In addition, the PLA also provides instruction type to FSM, enabling FSM to choose the next microstate properly

## Pin Descriptions

Pin Name	Type	Description
<b>A15 - A0</b>	<b>O/Z</b>	Parallel port address bus A15 (MSB) through A0 (LSB). The sixteen LSBs (A15–A0) are multiplexed to address external data/program memory or I/O. A15–A0 are placed in the high-impedance state in the hold mode.
<b>D15 - D0</b>	<b>I/O/Z</b>	Parallel port data bus D15 (MSB) through D0 (LSB). D15–D0 are multiplexed to transfer data between the core CPU and external data/program memory or I/O devices. D15–D0 are placed in the high-impedance state when not output or when RS_n or HOLD_n is asserted.
<b>IACK_n</b>	<b>O/Z</b>	Data, program, and I/O space select signals. Active low.
<b>INT0_n</b> <b>INT1_n</b> <b>INT2_n</b> <b>INT3_n</b>	<b>I</b>	External users interrupt inputs. INT0_n - INT3_n are prioritized and are maskable by the interrupt mask register and the interrupt mode bit. INT0_n - INT3_n can be polled and reset by the interrupt flag register.
<b>NMI_n</b>	<b>I</b>	Nonmaskable interrupt. NMI_n is an external interrupt that cannot be masked by way of the INTM or the IMR. When NMI_n is activated, the processor traps to the appropriate vector location.
<b>RS_n</b>	<b>I</b>	Reset input. RS_n causes the DSP5 to terminate execution and forces the program counter to 0FF80h. When RS_n is brought to a high level, execution begins at location 0FF80h of the program memory. RS_n affects various registers and status bits.
<b>MP/MC_n</b>	<b>I</b>	Microprocessor/microcomputer mode-select pin. If active-low at reset (microcomputer mode), MP/MC_n causes the internal program ROM to be mapped into the upper program memory space. In the microprocessor mode, off-chip memory and its corresponding addresses (instead of internal program ROM) are accessed by the DSP5.
<b>BIO_n</b>	<b>I</b>	Branch control input. A branch can be conditionally executed when BIO_n is active. If low, the processor executes the conditional instruction. The BIO_n condition is sampled during the decode phase of the pipeline for the XC instruction, and all other instructions sample BIO_n during the read phase of the pipeline.
<b>XF</b>	<b>O/Z</b>	External flag output (latched software-programmable signal). XF is set high by the SSBX XF instruction, set low by RSBX XF instruction or by loading the ST1 status register. XF is used for signaling other processors in multiprocessor configurations or as a general-purpose output pin.
<b>DS_n</b> <b>PS_n</b> <b>IS_n</b>	<b>O/Z</b>	Data, program, and I/O space select signals. DS_n, PS_n, and IS_n are always high unless driven low for communicating to a particular external space. Active period corresponds to valid address information. Placed into a high-impedance state in hold

		mode.
<b>MSTRB_n</b>	<b>O/Z</b>	Memory strobe signal. MSTRB_n is always high unless low-level asserted to indicate an external bus access to data or program memory. Placed in high-impedance state in hold mode.
<b>READY</b>	<b>I</b>	Data-ready input. READY indicates that an external device is prepared for a bus transaction to be completed. If the device is not ready (READY is low), the processor waits one cycle and checks READY again. Note that the processor performs ready-detection if at least two software wait states are programmed. The READY signal is not sampled until the completion of the software wait states.
<b>R/W_n</b>	<b>O/Z</b>	Read/write signal. R/W indicates transfer direction during communication to an external device and is normally high (in read mode), unless asserted low when the DSP performs a write operation. Placed in the high-impedance state in hold mode.
<b>IOSTRB_n</b>	<b>O/Z</b>	I/O strobe signal. IOSTRB_n is always high unless low level asserted to indicate an external bus access to an I/O device. Placed in high-impedance state in hold mode.
<b>HOLD_n</b>	<b>I</b>	Hold input. HOLD_n is asserted to request control of the address, data, and control lines. When acknowledged by the DSP5, these lines go into high-impedance state.
<b>HOLDA_n</b>	<b>O/Z</b>	Hold acknowledge signal. HOLDA_n indicates to the external circuitry that the processor is in a hold state and that the address, data, and control lines are in a high-impedance state, allowing them to be available to the external circuitry
<b>CLKOUT</b>	<b>O/Z</b>	Master clock output signal. CLKOUT cycles at the machine-cycle rate of the CPU. The internal machine cycle is bounded by the falling edges of this signal.
<b>CLKMD1</b> <b>CLKMD2</b> <b>CLKMD3</b>	<b>I</b>	Clock mode external/internal input signals. CLKMD1, CLKMD2, and CLKMD3 allow you to select and configure different clock modes, such as crystal, external clock, and various PLL factors.
<b>X2/CLKIN</b>	<b>I</b>	Input pin to internal oscillator from the crystal. If the internal (crystal) oscillator is not being used, a clock can become input to the device using this pin. The internal machine cycle time is determined by the clock operating-mode pins (CLKMD1, CLKMD2 and CLKMD3).
<b>X1</b>	<b>O</b>	Output pin from the internal oscillator for the crystal. If the internal oscillator is not used, X1 should be left unconnected.
<b>TOUT</b>	<b>O/Z</b>	Timer output. TOUT signals a pulse when the on-chip timer counts down past zero. The pulse is a CLKOUT-cycle wide.
<b>BCLKR0</b> <b>BCLKR1</b>	<b>I</b>	Receive clocks. External clock signal for clocking data from the data-receive (DR) pin into the buffered serial port receive shift registers (RSRs). Must be present during buffered serial port transfers. If the buffered serial port is not being used, BCLKR0 and BCLKR1 can be sampled as an input by way of IN0 bit of the SPC register.
<b>BCLKX0</b>	<b>I/O/Z</b>	Transmit clock. Clock signal for clocking data from the serial port transmit shift register

<b>BCLKX1</b>		(XSR) to the data transmit (DX) pin. BCLKX can be an input if MCM in the serial port control register is cleared to 0. It also can be driven by the device at $1/(CLKDV + 1)$ where CLKDV range is 0–31 CLKOUT frequency when MCM is set to 1. If the buffered serial port is not used, BCLKX can be sampled as an input by way of IN1 of the SPC register.
<b>BDR0</b> <b>BDR1</b>	<b>I</b>	Buffered serial-data-receive input. Serial data is received in the RSR by BDR0/BDR1.
<b>BDX0</b> <b>BDX1</b>	<b>O/Z</b>	Buffered serial-port-transmit output. Serial data is transmitted from the XSR by way of BDX. BDX0 and BDX1 are placed in the high-impedance state when not transmitting.
<b>BFSR0</b> <b>BFSR1</b>	<b>I</b>	Frame synchronization pulses for receive input. The falling edge of the BFSR pulse initiates the data-receive process, beginning the clocking of the RSR.
<b>BFSX0</b> <b>BFSX1</b>	<b>I/O/Z</b>	Frame synchronization pulses for transmit input/output. The falling edge of the BFSX pulse initiates the data-transmit process, beginning the clocking of the XSR. Following reset, the default operating condition of BFSX is an input. BFSX0 and BFSX1 can be selected by software to be an output when TXM in the serial control register is set to 1
<b>CLKR0</b> <b>CLKR1</b>	<b>I</b>	Receive clocks. External clock signal for clocking data from the data receive (DR) pin into the serial port receive shift register (RSR). Must be present during serial port transfers. If the serial port is not being used, CLKR0 and CLKR1 can be sampled as an input via IN0 bit of the SPC register.
<b>CLKX0</b> <b>CLKX1</b>	<b>I/O/Z</b>	Transmit clock. Clock signal for clocking data from the serial port transmit shift register (XSR) to the data transmit (DX) pin. CLKX can be an input if MCM in the serial port control register is cleared to 0. It also can be driven by the device at $1/4$ CLKOUT frequency when MCM is set to 1. If the serial port is not used, CLKX can be sampled as an input via IN1 of the SPC register.
<b>DR0</b> <b>DR1</b>	<b>I</b>	Serial-data-receive input. Serial data is received in the RSR by DR.
<b>DX0</b> <b>DX1</b>	<b>O/Z</b>	Serial port transmits output. Serial data is transmitted from the XSR via DX. DX0 and DX1 are placed in the high-impedance state when not transmitting.
<b>FSR0</b> <b>FSR1</b>	<b>I</b>	FSR1   Frame synchronization pulses for receive input. The falling edge of the FSR pulse initiates the data-receive process, beginning the clocking of the RSR.
<b>FSX0</b> <b>FSX1</b>	<b>I/O/Z</b>	Frame synchronization pulses for transmit input/output. The falling edge of the FSX pulse initiates the data transmit process, beginning the clocking of the XSR. Following reset, the default operating condition of FSX is an input. FSX0 and FSX1 can be selected by software to be an output when TXM in the serial control register is set to 1.
<b>TCLKR</b>	<b>I</b>	TDM receive clock input
<b>TDR</b>	<b>I</b>	TDM serial data-receive input
<b>TFSR/TADD</b>	<b>I/O</b>	TDM receive frame synchronization or TDM address

<b>TCLKX</b>	<b>I/O/Z</b>	TDM transmit clock
<b>TDX</b>	<b>O/Z</b>	TDM serial data-transmit output
<b>TFSX/TFRM</b>	<b>I/O/Z</b>	TDM transmit frame synchronization
<b>HD0-HD7</b>	<b>I/O/Z</b>	Parallel bi-directional data bus. HD0–HD7 are placed in the high-impedance state when not outputting data. These pins each have bus holders similar to those on the address/data bus, but which are always enabled.
<b>HCNTL0</b> <b>HCNTL1</b>	<b>I</b>	Control inputs
<b>HBIL</b>	<b>I</b>	Byte-identification input
<b>HCS_n</b>	<b>I</b>	Chip-select input
<b>HDS1_n</b> <b>HDS2_n</b>	<b>I</b>	Data strobe inputs
<b>HAS_n</b>	<b>I</b>	Address strobe input
<b>HR/W_n</b>	<b>I</b>	Read/write input
<b>HRDY</b>	<b>O/Z</b>	Ready output.
<b>NINT_n</b>	<b>O/Z</b>	Interrupt output. When the DSP is in reset, this signal is driven high.

# Instructions and Finite State Machine

## Type 1 : 1 cycle

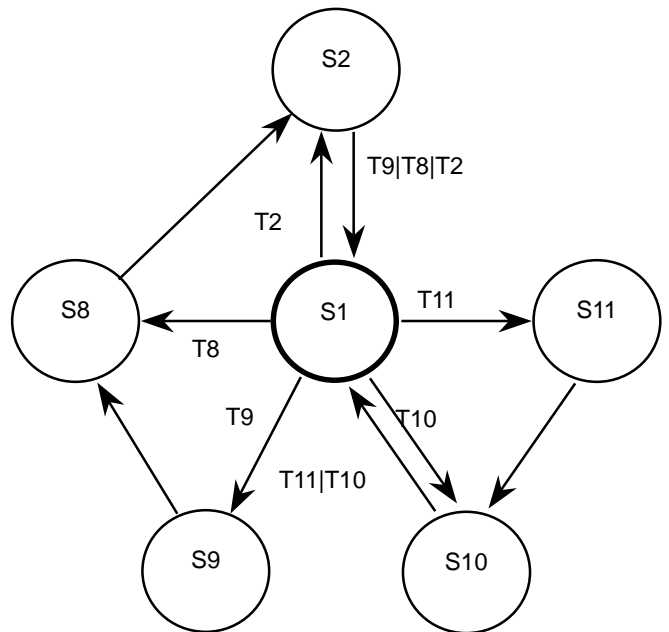
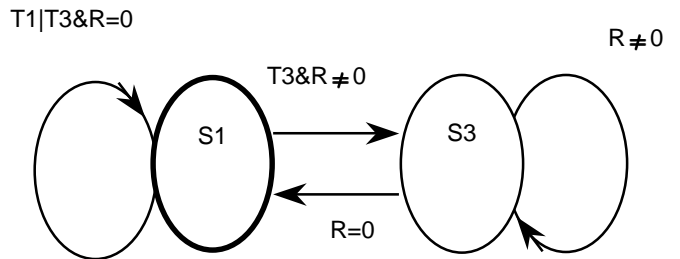
LDARP, LDK9DP, RETF, RND, RPTK, RSBX, SSBX, XC (8)

## Type 2 : 2 cycles

ADDM, ANDM, CMPR, DST, ORM, RPTBD, RPTLK, RPTZ, XORM (9)

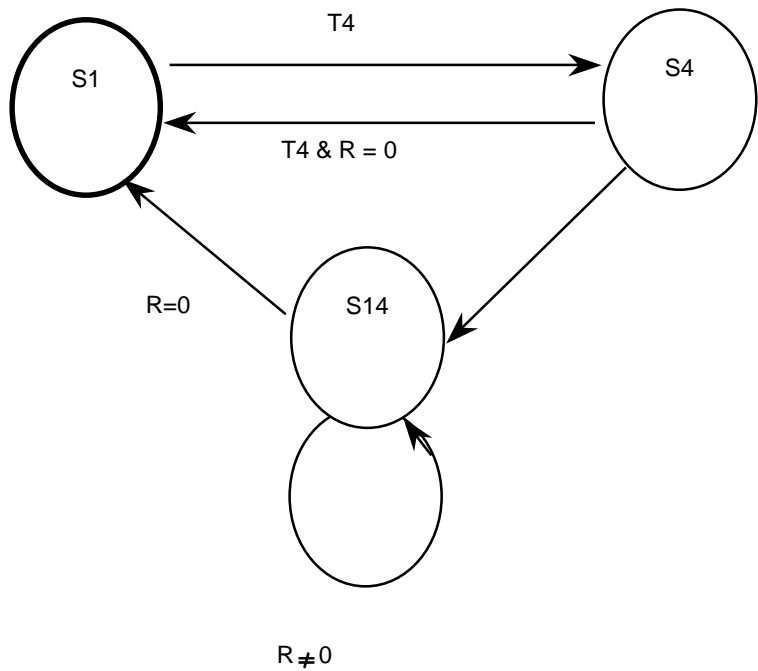
## Type 3 : n cycles

ABDST, ABS, ADD, ADDACC, ADDASM, ADDC, ADDH, ADDDUAL, ADDSFT, ADDTS, ADDS, AND, ANDSHIFT, BIT, BITT, CMPS, CMPL, DADD, DADST, DELAY, DLD, DRSUB, DSADT, DSUB, DSUBT, EXP, FRAME, LDACC, LD, LD16, LDACCASM, LDK, LDK5ASM, LDM, LDMAC, LDMACR, LDMAS, LDMASR, LDMASM, LDR, LDU, LDT, LDSFT, LDTS, LMS, LTD, MAC, MACA, MACAR, MACR, MACAT, MACART, MACDUAL, MACRDUAL, MACSU, MAS, MASA, MASR, MASAT, MASART, MASDUAL, MAR, MASRDUAL, MAX, MIN, MPY, MPYA, MPYATACC, MPYU, MPYDUAL, MPYR, MVDD, NEG, NOP, NORM, OR, POPD, ORSHIFT, POPM, POLY, PSHD, PSHM, ROL, ROLTC, ROR, SACCD, SAT, SFTA, SFTC, SFTL, STADD, STH, STL, STHASM, STHSFT, STLASM, STLD, STLDT, STLM, STLSFT, STMAC, STMACR, STMAS, STMASR, STMPY, STRCD, STSUB, STT, STTTRN, SUB, SUBACC, SUBASM, SUBB, SUBC, SUBDUAL, SUBH, SUBS, SUBSFT, SUBTS, SQDST, SQUR, SQURA, SQURACC, SQURS, XOR, XORSHIFT (126)



**Type 4 : n+1 cycles**

ADDLK, ADDLK16, ANDLK,  
 ANDLK16, BITF, CMPM,  
 LDLK, LDLK16, LKSHIF, T,  
 MACSLK, MACTLK, MPYSLK,  
 MPYTLK, MVDK, MVDM,  
 MVDP, MVKD, MVMD, ORLK,  
 OILK16, PORTR, PORTW,  
 SRCCD, SUBLK, SUBLK16,  
 STLK, STM, XORLK, XORLK16 (29)



**Type 5 : n+2 cycles**

FIRS, MACD, MACP, MVPD (4)

**Type 6 : 3+n+1 cycles**

READA, WRITEA (2)

**Type 8 : 3 cycles**

LDDP, MVMM, RPT (3)

**Type 9 : 4 cycles**

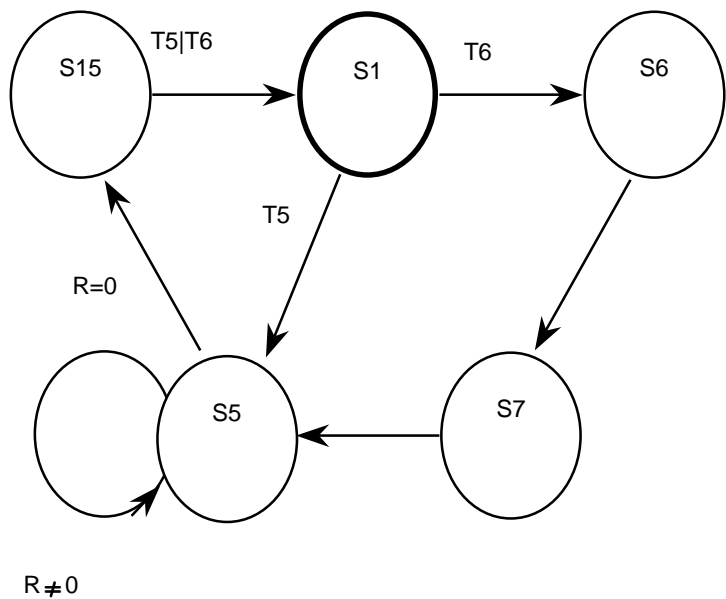
RPTB, IDLE (2)

**Type 10 : 4(True)/2(False) cycles**

B, BANZ, CALL (3)

**Type 11 : 5(True)/3(False) cycles**

BACC, BC, CALA, CC, INTR, RC, RESET,  
 RETE, TRAP (9)



## Verification Methods

- Model Simulation using ASM and C programs
- Gate Level Simulation using ASM and C programs
- FPGA Board Test using Commercial Application Programs
- Silicon Test using Commercial Application Programs

## Deliverables

- VHDL Source Code
- Post-Synthesis EDIF Netlist
- Test Bench and Test Vector (VHDL, Verilog)
- Simulation Scripts (Cadence NC-SIM, Verilog-XL)
- Synthesis Scripts (Synopsys Design Compiler)
- Documentation

## Contact Information

Infinior MicroSystems  
5F Jeongam Bldg, 769-12, Yeoksam2-Dong, Gangnam-Gu,  
Seoul, Korea 135-928

Phone : +82-2-6202-5500  
FAX : +82-2-6202-5555  
E-Mail : [sales@infinior.com](mailto:sales@infinior.com)  
URL : [www.infinior.com](http://www.infinior.com)

Copyright© 2000-2003 Infinior Microsystems, All Rights Reserved.

No part of this document may be reproduced in any form or means  
without the prior written permission of Infinior Microsystems.