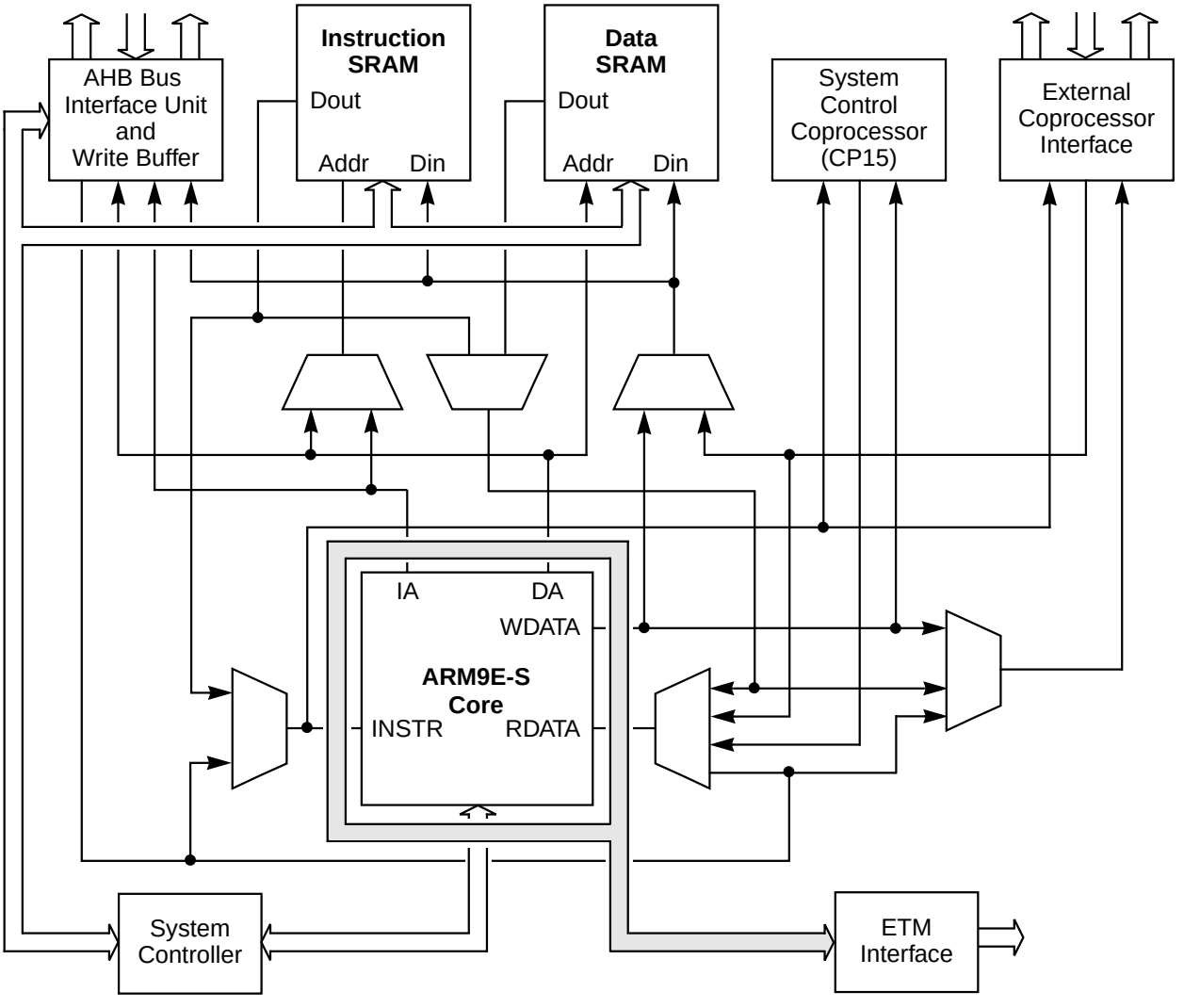
CW001105 ARM966E-S Microprocessor Core

Preliminary Datasheet

The CW001105 ARM966E-S core integrates the ARM9E-S 32-bit processor, an instruction RAM, a data RAM, a write buffer, and an AHB bus interface. The CW001105 supports both the 32-bit ARM and 16-bit Thumb instruction sets, allowing you to trade off between high performance and high code density. Additionally the CW001105 supports the ARM9E instruction extensions. It provides an enhanced multiplier for increased DSP performance. The CW001105 core is developed using LSI Logic's G12®-p performance process.

Figure 1 CW001105 Block Diagram



The AHB bus interface eases connection to cached and SRAM-based memory systems.

The CW001105 supports the ARM debug architecture and includes logic to assist in both hardware and software debug. It supports non-stopping hardware debug, which allows critical exception handlers to execute while debugging the system. The CW001105 provides real-time trace and supports external coprocessors.

Features

This section lists the key features of the CW001105 microprocessor core:

- ARM9E-S processor core
- Instruction and Data RAMs with independent sizes up to 512 Kbytes
- DMA Interface to Data RAM
- ARM Advanced High-performance Bus (AHB) interface unit with Write Buffer
  - 16-word Write Buffer depth at up to four addresses
  - Burst transfer generation
  - Support for split transactions
- External coprocessor interface
- System controller arbitrates between instruction and data memories and AHB
- Optional embedded trace module (ETM) provides real-time trace capability
- G12-p performance process
- 0.18-micron drawn gate length (0.13 effective channel length)
- System clock operates at 195 MHz tested under worst-case conditions, 1.62 V, 115 ˚C

Functional Description

This section briefly describes the main functional blocks of the CW001105.

ARM9E-S Processor Core

The ARM9E-S processor core has a Harvard bus architecture with separate instruction and data interfaces. This design allows concurrent instruction and data accesses, and greatly reduces the cycles per instruction of the processor. For optimal performance, single cycle memory accesses for both interfaces are required, although the core can be stalled for non-sequential accesses, or slower memory systems.

The processor is implemented using a five-stage pipeline:

- Instruction Fetch (F)
- Instruction Decode (D)
- Execute (E)
- Data Memory Access (M)
- Register Write (W)

ARM implementations are fully interlocked, so that software functions identically across different implementations without concern for pipeline effects.

System Controller

The system controller oversees the interactions between the Instruction RAM, Data RAM, and the Bus Interface Unit. It controls internal arbitration between the blocks and stalls the appropriate blocks when required.

CP15 System Control Coprocessor

The processor core uses a set of registers in the CP15 Coprocessor to control the functionality of the RAMs and the Write Buffer. These registers are accessed using the coprocessor instructions MCR and MRC.

Address Decoders

The address decoders determine whether a memory request accesses the internal RAM or the AHB interface. The address decoders provide a hit/miss indication to the system controller, which then either stalls the core if an AHB read or unbuffered write access is required or allows execution to continue if the access hits the RAM or is a buffered write.

Instruction and Data RAMs

The CW001105 incorporates internal instruction and data memories to allow high-speed operation without incurring the performance penalties of accessing the system bus. Typically, the CW001105 offers lower power solutions than cached alternatives because memory is segmented to conserve power. The Instruction and Data RAMs each consist of blocks of ASIC library compiled RAM. Logically, the RAM sizes can be of any size up to 64 Mbytes, but the practical limit is approximately 512 Kbytes for the G12 technology. The instruction and data memories can have unique sizes.

DMA Interface

The Direct Memory Access (DMA) interface allows an external device direct access to the CW001105 Data RAM. If a single-port Data RAM is used, then the DMA interface stalls the CW001105 microprocessor core during the DMA transfer. If a dual-port Data RAM is used, then the DMA interface does not stall the CW001105 during the DMA transfer.

AHB Interface Unit and Write Buffer

The AHB (Advanced High-performance Bus) is a new generation of AMBA bus, which meets the requirements of high-performance synthesizable designs. The AHB Interface Unit arbitrates between the external bus transaction sources within the CW001105. It stalls all other accesses until the current request has been completed. The AHB Interface Unit supports the following types of transactions: burst transfers, split transactions, single-cycle bus master handovers, single clock edge operations, and non-3-state implementations.

The Write Buffer is a 12-entry FIFO. It increases system performance.

External Coprocessor Interface

The CW001105 supports the connection of coprocessors through the external coprocessor interface. All types of ARM coprocessor instructions are supported. Coprocessors determine the instructions they need to execute using a pipeline follower in the coprocessor.

JTAG and Debug Port

The CW001105 debug interface is based on IEEE Std. 1149.1-1990. It allows the processor core to be stopped on a given instruction fetch (breakpoint), data access (watchpoint), or external debug request. The JTAG-style serial interface allows instructions to be serially inserted into the pipeline of the core without using the external data bus.

Embedded Trace Module Interface

This interface connects to an external Embedded Trace Module (ETM). The ETM provides a high-speed port for tracing of the processor core in real time.

Enhanced Instruction Set Summary

Table 1 lists the instruction enhancements made to the ARM9E-S instruction set. The ARM9E extensions improve the ARM architecture's performance in signal processing algorithms.

Table 1 ARM9E Instruction Set Summary

Instruction	Description	Instruction	Description
CLZ	Count Leading Zeros	QADD	Saturating Add
QDADD	Saturated Double Rn and Saturated Add	QDSUB	Saturated Double Rn and Saturated Subtract
QSUB	Saturating Subtract	SMLAxy	Signed Integer Multiply-Accumulate
SMLALxy	Signed Multiply-Accumulate	SMLAWy	Signed Integer Multiply-Accumulate
SMULxy	Signed Integer Multiply	SMULWy	Signed Integer Multiply

Signal Descriptions

This section lists the signals that make up the external interface to the CW001105. In the descriptions that follow, the verb *assert* means to drive TRUE or active. The verb *deassert* means to drive FALSE or inactive.

AHB Interface

HADDR[31:0] Address Bus Out **Output**

The CW001105 drives the AHB address on HADDR[31:0].

HBURST[2:0] Burst Type **Output**

This output indicates if the transfer forms part of a burst. Both 4-beat and 8-beat bursts are supported, where a beat is a clock tick. The CW001105 generates only single word transfers or bursts of unspecified, 4, or 8 word lengths to drain Store Multiples (STMs) posted in the write buffer.

HBURST[2:0]	Burst Type	Description
000	SINGLE	Single transfer
001	INCR	Incrementing burst of unspecified length
010	WRAP4	4-beat wrapping burst
011	INCR4	4-beat incrementing burst
100	WRAP8	8-beat wrapping burst
101	INCR8	8-beat incrementing burst
110	WRAP16	16-beat wrapping burst
111	INCR16	16-beat incrementing burst

HBUSREQ Bus Request **Output**

HBUSREQ is a signal from the CW001105 core to the bus arbiter. A HIGH in this output indicates that the core requires the bus.

HGRANT Bus Grant **Input**

A HIGH on this signal indicates the CW001105 core is currently the highest priority master. Ownership of the address/control signals changes at the end of a transfer when HREADY is HIGH. A master gets access to the bus when both HREADY and HGRANT are HIGH.

HLOCK **Locked Transfer** **Output**

When HLOCK is HIGH, the ARM9E-S core requires locked access to the bus, and no other masters should be granted the bus until HLOCK is LOW. HLOCK is asserted when the CW001105 is executing the SWAP instruction.

HPROT[3:0] **Protection Control** **Output**

This output provides additional information about a bus access. HPROT[3:0] are primarily intended for use by any module that implements some level of protection.

The signals indicate whether the transfer is an opcode fetch or data access, as well as if the transfer is a supervisor mode access or user mode access.

HPROT3 Cacheable	HPROT2 Bufferable	HPROT1 Supervisor	HPROT0 Data/Opcode	Description
–	–	–	0	Opcode Fetch
–	–	–	1	Data Access
–	–	0	–	User Access
–	–	1	–	Supervisor Access
–	0	–	–	Not Bufferable
–	1	–	–	Bufferable
0	–	–	–	Not Cacheable

Note that for the CW001105, HPROT3 is forced LOW (non-cacheable).

HRDATA[31:0] **Read Data Bus** **Input**

This bus transfers data from the bus slaves to the CW001105 core during read operations. The CW001105 core has a 32-bit wide data bus. The width can be easily extended outside the core to allow for higher bandwidth operation.

HREADY **Transfer Done In** **Input**

When HIGH, the HREADY signal indicates the transfer on the bus has finished. Drive this signal LOW to extend a transfer.

Note: Slaves on the bus require HREADY to be both an input and an output.

HRESETn **Reset** **Input**

This input is the active-LOW system reset.

HRESP[1:0] **Transfer Response In** **Input**

HRESP[1:0] provide additional information on the status of a transfer. When a slave must insert a number of wait states prior to decoding what response is to be given, then it must drive the response to OKAY.

HRESP[1:0]	Transfer Response	Description
00	OKAY	When HREADY is HIGH, the transfer has completed.
01	ERROR	This response shows an error has occurred. The error condition needs to be signaled to the bus master to make it aware that the transfer is unsuccessful. A 2-cycle response is required for an Error condition.
10	RETRY	The Retry response shows the transfer has not yet completed, so the bus master needs to retry the transfer. The master continues to retry the transfer until the transfer completes. A 2-cycle RETRY response is required.
11	SPLIT	The Split response indicates the transfer has not completed successfully. The bus master must retry the transfer when it is next granted access to the bus. The slave requests access to the bus on behalf of the master when the transfer can complete. A 2-cycle SPLIT response is required.

HSIZE[2:0] **Transfer Size** **Output**

This output indicates the size of the transfer, which is typically byte (8 bits), halfword (16 bits), or word (32 bits). The bus protocol allows for transfer sizes up to 1024 bits.

HSIZE[2:0]	Transfer Size	Description
000	8 bits	Byte
001	16 bits	Halfword
010	32 bits	Word
011	64 bits	2-word line
100	128 bits	4-word line
101	256 bits	8-word line
110	512 bits	
111	1024 bits	

HTRANS[1:0] **Transfer Type Out** **Output**

This output indicates the type of the current transfer.

HTRANS[1:0]	Transfer Type	Description
00	Idle	No data transfer required.
01	Busy	Used to insert an idle cycle in the middle of a burst of transfers.
10	Non-sequential	Indicates first transfer of a burst or single transfer.
11	Sequential	The control information is identical to the previous transfer. The address is equal to the address of the previous transfer plus the size (in bytes). For wrapping bursts, the address of the transfer wraps at the address boundary equal to the size (in bytes) multiplied by the number of beats in the transfer (either 4, 8, or 16).

HWDATA[31:0]

Write Data Bus **Output**

This bus transfers data from the master to the bus slaves during write operations. The width easily can be extended external to the core to allow for higher bandwidth operation.

HWRITE **Transfer Direction Out** **Output**

When HWRITE is HIGH, the transfer is a write. When HWRITE is LOW, the transfer is a read.

Coprocessor Interface

CHSDE[1:0] **Coprocessor Handshake Decode** **Input**

These inputs are the handshake signals from the decode stage of the coprocessor's pipeline follower.

CHSDE[1:0]	Encoding
10	ABSENT
00	WAIT
01	GO
11	LAST

CHSEX[1:0] **Coprocessor Handshake Execute** **Input**

These inputs are the handshake signals from the execute stage of the coprocessor's pipeline follower.

CHSEX[1:0]	Encoding
10	ABSENT
00	WAIT
01	GO
11	LAST

CPCLKEN **Coprocessor Clock Enable** **Output**

This clock enable controls the timing of the coprocessor interface. It is used in conjunction with CLK to effectively run the coprocessor at a higher frequency than the data bus.

CPDIN[31:0] **Coprocessor Data In** **Input**

This 32-bit bus is the coprocessor data bus for transferring MRC and STC data from the coprocessor to the CW001105.

CPDOUT[31:0]

Coprocessor Data Out **Output**

This 32-bit bus is the coprocessor data bus for transferring data to the coprocessor.

CPINSTR[31:0]

Coprocessor Instruction **Output**

This 32-bit bus is the coprocessor instruction data bus for transferring instructions to the pipeline follower in the coprocessor.

CPLATECANCEL

Coprocessor Late Cancel **Output**

When CPLATECANCEL is HIGH during the first memory cycle of a coprocessor instruction execution, the coprocessor instruction must be cancelled without updating any internal state.

This signal is asserted only in cycles where the previous instruction accessed memory and a data abort occurred.

CPPASS **Coprocessor Pass** **Output**
A HIGH on this signal indicates that there is a coprocessor instruction in the execute stage of the pipeline that needs to be executed.

CPTBIT **Coprocessor Interface in Thumb State** **Output**
When CPTBIT is HIGH, the coprocessor interface is in Thumb state (16-bit instructions); otherwise the interface supports 32-bit instruction execution.

nCPMREQ **Not Coprocessor Memory Request** **Output**
When nCPMREQ is LOW on a rising CLK edge and CPCLKEN is HIGH, the instruction on CPINSTR must enter the coprocessor pipeline follower's decode stage, and the instruction previously in the pipeline follower's decode stage should enter its execute stage.

nCPTRANS **Not Coprocessor Translate** **Output**
When nCPTRANS is LOW, the coprocessor interface is in a non-privileged state. When nCPTRANS is HIGH, the coprocessor interface is in a privileged state. The coprocessor should sample this signal on every cycle when determining the coprocessor response.

Instruction RAM Signals

IADDR[23:0] **Instruction RAM Address** **Output**
This 24-bit bus contains the Instruction RAM address. Addressing is performed on word boundaries.

IENABLE **Word-Based Instruction Chip Enable** **Output**
Assertion HIGH of this output indicates the Instruction RAM data bus is enabled.

IRDATA[31:0]
Instruction RAM Read Data **Input**
Data to be read from the Instruction RAM is placed on this 32-bit bus.

IWDATA[31:0]
Instruction RAM Write Data **Output**
This 32-bit bus contains write data to the Instruction RAM.

IWE[3:0] **Byte-Based Instruction Write Enable** **Output**
Assertion HIGH of this output enables writes to the Instruction RAM.

IWE Bit	Function
IWE3	Write Enable for IWDATA[31:24]
IWE2	Write Enable for IWDATA[23:16]
IWE1	Write Enable for IWDATA[15:8]
IWE0	Write Enable for IWDATA[7:0]

Data RAM Signals

The Data RAM interface supports both single-port and dual-port RAMs. When single-port RAMs are used, the CW001105 stalls during DMA transfers. When dual-port RAMs are used, the CW001105 does not need to be stalled during DMA transfers.

The CW001105 uses the signals described below to access the Data RAM for both single-port and dual-port RAM implementations. The DMA interface shares these signals with the CW001105 for single-port RAM implementations.

DADDR[23:0] **Data RAM Address** **Output**
This 24-bit bus contains the Data RAM address. Addressing is performed on word boundaries.

DENABLE **Word-Based Data Chip Enable** **Output**
Assertion HIGH of this output indicates the Data RAM data bus is enabled.

DRDATA[31:0]
Data RAM Read DataInput
This 32-bit bus contains data read from the Data RAM.

DWDATA[31:0]
Data RAM Write DataOutput
This 32-bit bus provides write data to the Data RAM.

DWE[3:0] **Byte-Based Data Write Enable** **Output**
Assertion HIGH of this output enables writes to the Data RAM.

DWE Bit	Function
DWE3	Write Enable for DWDATA[31:24]
DWE2	Write Enable for DWDATA[23:16]
DWE1	Write Enable for DWDATA[15:8]
DWE0	Write Enable for DWDATA[7:0]

The DMA Interface uses the following signals to access only the second port of a dual-port RAM.

DADDR2[23:0]
Data RAM Address **Output**
This 24-bit bus contains the Data RAM address. Addressing is performed on word boundaries.

DENABLE2 **Word-Based Data Chip Enable** **Output**
Assertion HIGH of this output indicates the Data RAM data bus is enabled.

DRDATA2[31:0]
Data RAM Read Data **Input**
This 32-bit bus contains data read from the Data RAM.

DWDATA2[31:0]
Data RAM Write Data **Output**
This 32-bit bus provides write data to the Data RAM.

DWE2[3:0] **Byte-Based Data Write Enable** **Output**
Assertion HIGH of this output enables writes to the Data RAM.

DWE2 Bit	Function
DWE3	Write Enable for DWDATA[31:24]
DWE2	Write Enable for DWDATA[23:16]
DWE1	Write Enable for DWDATA[15:8]
DWE0	Write Enable for DWDATA[7:0]

DMA Signals

DMAA[25:0] **DMA Address** **Input**

This 26-bit address contains the byte address for DMA transfers. Tie all unused address bits LOW.

DMAD[31:0] **DMA Write Data** **Input**

This 32-bit bus contains the DMA write data to the Data RAM.

DMAENABLE **DMA Port Enable** **Input**

DMAENABLE must be asserted HIGH for a DMA transfer to proceed. Assert DMAENABLE LOW to save power when the DMA interface is not being used. Tie DMAENABLE LOW when the DMA interface is not used in the implementation.

DMAMAS[1:0]

DMA Memory Access Size **Input**

DMAMAS[1:0] encodes the size of DMA writes. DMA reads are always one word wide.

DMAMAS[1:0]	Memory Access Size
00	Byte
01	Halfword
10	Word
11	Reserved

DMAnREQ **DMA Request** **Input**

DMAnREQ is an active-LOW DMA transfer request. Tie this input HIGH when the DMA interface is not used.

DMAnRW **DMA Write Not Read** **Input**

DMAnRW is the DMA Read/Write signal.

DMAnRW	Function
0	Read
1	Write

DMARData[31:0]

DMA Read Data **Output**

This 32-bit bus contains DMA Data read from the Data RAM.

DMAReady **DMA Ready** **Output**
DMAReady is asserted HIGH when the CW001105 is stalled due to a DMA Wait request. DMAReady must be sampled HIGH before a DMA transfer to/from a single-port Data RAM can take place.

DMAWait **DMA Wait Request** **Input**
DMAWait is asserted HIGH to stall the CW001105 before proceeding with a DMA transfer to/from a single-port Data RAM implementation. A HIGH on DMAReady indicates when the CW001105 is stalled.

Only use this signal for single-port Data RAM implementations. Tie it LOW for dual-port Data RAM implementations.

Debug Signals

COMMRX **Communications Channel Receive** **Output**
When HIGH, this signal indicates that the Comms Channel Receive Buffer has data that the ARM9E-S CPU can read.

COMMTX **Communications Channel Transmit** **Output**
When HIGH, this signal indicates the Comms Channel Transmit Buffer is empty.

DBGACK **Debug Acknowledge** **Output**
When HIGH, DBGACK indicates that the ARM9E-S is in debug mode.

DBGDEWPT **Debug Watchpoint** **Input**
This input halts the processor for debug purposes. If HIGH at the end of a data memory request cycle, this input causes the ARM9E-S processor core to enter the debug state.

DBGEN **Debug Enable** **Input**
A LOW on this input disables the debug features of the CW001105. Tie this input LOW when debugging is not required.

DBGEXT[1:0] **Breakpoint/Watchpoint External Condition** **Input**
These inputs to the EmbeddedICE logic allow breakpoints/watchpoints to be dependent on external conditions.

DBGIEBKPT **Processor Execution Breakpoint** **Input**

Assertion HIGH of this input halts processor execution for debug purposes. If DBGIEBKPT is HIGH at the end of an instruction fetch, then the ARM9E-S processor core enters the debug state if that instruction reaches the execute stage of the processor's pipeline.

DBGINSTREXEC

Instruction Executed **Output**

A HIGH assertion of this output indicates that the instruction in the execute stage of the processor's pipeline has been executed.

DBGIR[3:0] **TAP Controller Instruction Register** **Output**

These outputs reflect the current instruction loaded into the TAP controller instruction register. They change when the TAP state machine is in the UPDATE_IR state on the rising edge of CLK when DBGTCKEN is asserted.

DBGnTDOEN **DBGTDO 3-State Enable** **Output**

When LOW, this signal indicates there is serial data on the DBGTDO output. DBGnTDOEN can be used as part of the output enable on a packaged part's DBGTDO pin.

DBGnTRST **Not Test Reset** **Input**

This active-LOW input is the internally synchronized reset signal for the EmbeddedICE internal state.

DBGRNG[1:0] **Watchpoint Register Match** **Output**

These outputs indicate that the corresponding EmbeddedICE Watchpoint register has matched the conditions currently present on the address, data, and control buses. These signals are independent of the state of the watchpoint's enable control bit.

DBGRQI **Internal Debug Request** **Output**

This signal is the debug request signal presented to the processor core's debug logic. It is the ANDing of EDBGRQ as presented to the CW001105 and bit 1 of the Debug Control Register.

DBGSCREG[4:0]

Scan Chain Register **Output**

These outputs reflect the ID number of the scan chain currently selected by the TAP controller. They change

when the TAP state machine is in the UPDATE_DP state on the rising edge of CLK when DBGTCKEN is asserted.

DBGSDIN **Boundary Scan Serial Input Data** **Output**
This output contains the serial data to be applied to an external scan chain.

DBGSDOUT **Boundary Scan Serial Output Data** **Input**
DBGSDOUT is the serial data input from an external scan chain. When an external scan chain is not implemented, tie this signal LOW.

DBGTAPSM[3:0]
TAP Controller State Machine **Output**
This bus reflects the current state of the TAP controller state machine. The TAP controller follows the IEEE 1149.1 Test Access Port protocol.

DBGTCKEN **Test Clock Enable** **Input**
This input is the synchronous enable for the test clock.

DBGTDI **Test Data In** **Input**
DBGTDI contains data input from the boundary scan logic.

DBGTDO **Test Data Out** **Output**
The CW001105 outputs test data on DBGTDO from its boundary scan logic.

DBGTMS **Test Mode Select** **Input**
DBGTMS is the JTAG test mode select signal. The test mode follows the IEEE 1149.1 Test Access Port protocol.

EDBGRQ **External Debug Request** **Input**
An external debugger asserts this signal HIGH to force the processor to enter the debug state.

ETM Interface Signals

These signals are part of the Trace module interface. All ETM outputs are registered from the corresponding core internal signals.

ETMBIGEND **Endian Mode** **Output**
This output indicates the endian mode for the ETM. When this signal is HIGH, the mode is big endian; when ETMBIGEND is LOW, the mode is little endian.

ETMCHSD[1:0]

ETM Coprocessor Handshake Decode **Output**

These outputs are the handshake signals from the decode stage of the coprocessor's pipeline follower.

ETMCHSD[1:0]	Encoding
10	ABSENT
00	WAIT
01	GO
11	LAST

ETMCHSE[1:0]

ETM Coprocessor Handshake Execute **Output**

These outputs are the handshake signals from the execute stage of the coprocessor's pipeline follower.

ETMCHSE[1:0]	Encoding
10	ABSENT
00	WAIT
01	GO
11	LAST

ETMDA[31:0] **ETM Data Address** **Output**

This 32-bit bus contains the ETM data address.

ETMDABORT **ETM Data Abort** **Output**

Assertion of this signal indicates a data abort to the ARM9E-S core.

ETMDBGACK **ETM Debug Mode Indication** **Output**

When HIGH, this signal indicates that the processor is in debug state.

ETMDMAS[1:0]

ETM Data Size IndicatorOutput

These signals indicate the data size of the ETM. They become valid in the same cycle as the data address bus.

DMAS[1:0]	Transfer Size
00	Byte
01	Halfword
10	Word
11	Reserved

ETMDMORE **ETM Sequential Data Indication** **Output**

The ETMDMORE signal is active during load and store multiple instructions and only goes HIGH when ETMDnMREQ is LOW. This signal effectively gives the same information as ETMDSEQ, but a cycle ahead. This information is provided to allow external logic more time to decode sequential cycles.

ETMDnMREQ **ETM Data Memory Request** **Output**

This signal is asserted HIGH when the CW001105 is making a request to ETM data memory.

ETMDnRW **ETM Data R/W** **Output**

If this signal is LOW at the end of the cycle, then any data memory access in the following cycle is a read; if this signal is HIGH, then the access is a write.

ETMDSEQ **ETM Sequential Data Indication** **Output**

If this signal is HIGH at the end of the cycle, then any data memory access in the following cycle is sequential from the last data memory access.

ETMEN **ETM Enable** **Input**

When this signal is HIGH, the ETM is enabled and the ARM9E-S interface signals are driven out of this module, pipelined by one clock stage.

ETMHIVECS **Exception Vector Location** **Output**

When this output is LOW, the ARM9E-S exception vectors start at address 0x0000.0000. When this signal is HIGH, the ARM9E-S exception vectors start at address 0xFFFF.0000. This output is a static configuration signal.

ETMIA[31:1] **ETM Instruction Address Bus** **Output**

This 31-bit bus contains the address for the ETM.

ETMID31To25[31:25]

Bits [31:25] of the Instruction Data **Output**

These outputs reflect the status of bits [31:25] of the instruction data read by the CW001105.

ETMID15To11[15:11]

Bits [15:11] of the Instruction Data **Output**

These outputs reflect the status of bits [15:11] of the instruction data read by the CW001105.

ETMInMREQ **ETM Instruction Memory Request** **Output**
The CW001105 drives this output LOW to indicate that an instruction fetch is occurring.

ETMINSTREXEC
ETM Instruction Execute Indicator **Output**
Assertion HIGH of this output indicates that the instruction in the execute stage of the processor pipeline has been executed.

ETMINSTRVALID
ETM Instruction Valid **Output**
Assertion HIGH of this output indicates that the current instruction is valid for the ETM.

ETMISEQ **ETM Sequential Instruction** **Output**
The ETMISEQ signal indicates whether the fetch is sequential (HIGH) or non-sequential (LOW) to the previous access.

ETMITBIT **ETM Thumb Indication** **Output**
When this signal is LOW, the processor is in ARM state, and 32-bit instructions are fetched. When ETMITBIT is HIGH, the processor is in Thumb state, and 16-bit instructions are fetched.

ETMLATECANCEL
ETM Coprocessor Late Cancel Indicator **Output**
A HIGH on this output during the first memory cycle of a coprocessor instruction informs the coprocessor to cancel the instruction without changing any internal state. This signal is only asserted in cycles where the previous instruction accessed memory and a data abort occurred.

ETMnWAIT **ETM Clock Stall** **Output**
Driving this output LOW stalls the ETM.

ETMPASS **ETM Coprocessor Instruction Execute Indicator** **Output**
A HIGH on this signal indicates that there is a coprocessor instruction in the execute stage of the pipeline, which needs to be executed.

ETMPROCID[31:0]
ETM Process ID **Output**
This 32-bit output contains the Process ID for the ETM.

ETMPROCIDWR

ETM Process ID Write **Output**

This output is asserted HIGH when ETMPROCID is written.

ETMRDATA[31:0]

ETM Read Data **Output**

This 32-bit bus contains ETM read data.

ETMRNGOUT[1:0]

ETM Watchpoint Register Match **Output**

This output indicates that corresponding EmbeddedICE watchpoint register has matched the conditions currently present on the address, data, and control buses. This signal is independent of the state of the watchpoint's enable control bit.

ETMWDATA[31:0]

ETM Write Data **Output**

This 32-bit bus contains ETM write data.

FIFOFULL **ETM FIFO FULL** **Input**

This input is asserted HIGH when the ETM FIFO is full. Tie this signal LOW when an ETM is not used.

TAPID[31:0] **Boundary Scan ID Code** **Input**

This bus specifies the ID Code value shifted out on DBGTDO when the IDCODE instruction is entered into the TAP Controller.

Miscellaneous Signals

BIGENDOUT **Big Endian** **Output**

When this output is HIGH, the CW001105 is in big-endian mode (byte 0 is the most-significant bit). When this output is LOW, the CW001105 is in little-endian mode.

This input is a static configuration signal. It must remain at one value from reset or be changed using a carefully constructed code sequence to avoid software problems.

CLK **System Clock** **Input**

CLK is the CW001105 system clock. CLK can be stretched in either state (held HIGH or LOW).

HCLKEN **HCLK Enable** **Input**

HCLKEN is used in conjunction with CLK to effectively run the CW001105 at a higher frequency than the AHB system bus. HCLKEN is HIGH for a single CLK period and signifies the rising edge of the AHB clock HCLK. All AHB outputs transition on the CLK rising edge in which HCLKEN is asserted.

nFIQ **Not Fast Interrupt** **Input**

This active-LOW input is the ARM Fast interrupt request. The CW001105 supports synchronous interrupts only.

nIRQ **Not Interrupt Request** **Input**

This active-LOW input is the ARM interrupt request. The CW001105 supports synchronous interrupts only.

Initialization Control Signals

INITRAM **RAM Enable Configuration** **Input**

When INITRAM is HIGH, the Instruction and Data RAMs are enabled at the end of reset. When INITRAM is LOW, the Instruction and Data RAMs are disabled coming out of reset.

This input is a static configuration signal. Its value is sampled at reset only.

VINITHI **High Vectors Configuration** **Input**

When VINITHI is LOW at reset, the ARM9 exception vectors start at address 0x0000.0000. When VINITHI is HIGH, the exception vectors start at 0xFFFF.0000.

This signal is a static configuration signal. Its value is sampled at reset only.

ATPG Scan Control Signals

This section describes the automatic test pattern generation (ATPG) scan control signals.

SCANEN **Scan Enable** **Input**

Assertion HIGH of this input enables the scanning of data through the scan chain.

SI0 **Scan Chain In** **Input**

SI0 is the input for serial scan chain 0.

SI1	**Scan Chain In** SI1 is the input for serial scan chain 1.	**Input**
SI2	**Scan Chain In** SI2 is the input for serial scan chain 2.	**Input**
SI3	**Scan Chain In** SI3 is the input for serial scan chain 3.	**Input**
SO0	**Scan Chain Out** SO0 is the output for serial scan chain 0.	**Output**
SO1	**Scan Chain Out** SO1 is the output for serial scan chain 1.	**Output**
SO2	**Scan Chain Out** SO2 is the output for serial scan chain 2.	**Output**
SO3	**Scan Chain Out** SO3 is the output for serial scan chain 3.	**Output**

Physical Specifications

The CW001105 core has a single 1x clock input. Clock duty cycles can vary from 30% to 70% at maximum frequency. Variance is greater at lower frequencies.

The CW001105 operates at 195.16 MHz tested under worst-case conditions: 1.62 V, 115 ˚C.

Table 2 shows the size of the CW001105 in G12-p 3+2 layer metal technology.

Table 2 CW001105 Physical Layout Size

Parameter	Size
Technology	G12-p
Width	1.53 mm
Height	1.53 mm

AC Timing

All AC timing values are relative to the system clock (CLK) input to the CW001105.

Input setup time is measured from the time the signal is valid to the rising edge of CLK. Input hold time is measured from the rising edge of CLK to the time the signal goes invalid. For input setup times, the driver must drive the signal valid before any receivers need it. For input hold times, the driver must hold the signal valid longer than needed by any receiver. The maximum and minimum delay times for outputs are measured from the rising edge of CLK to the time the signal is valid.

Figure 2 shows how AC timing is measured. Table 3 shows the CW001105 timing conditions.

Figure 2 AC Specifications

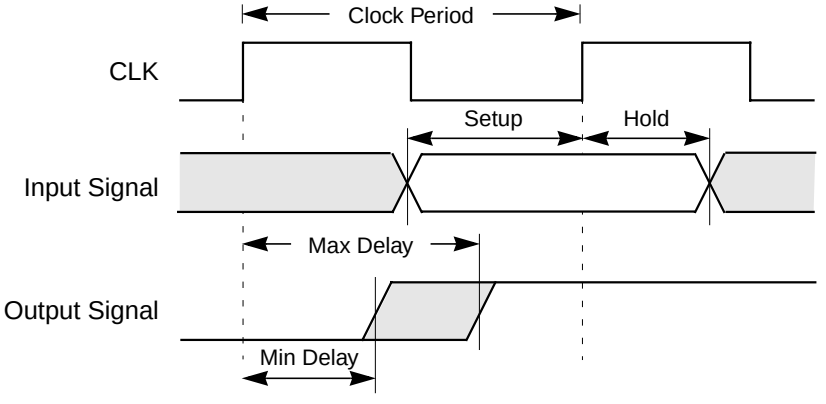


Table 3 CW001105 Core Timing Conditions for G12-p Process

AC Timing	Process	V_{DD} (V)	Junction Temperature (˚C)
lsi_bc	0.80	1.89	–40
lsi_wc10	1.17	1.62	115

AC Timing Parameters

This section describes various parameters related to the AC timing tables below.

Operating Conditions

The core is characterized for:

- Setup times: lsi_wc10
- Hold times: lsi_bc

Note: Be sure to specify operating frequency in the *design*.`lsitkdelspec` file. If the frequency is not specified, the `lsidelay` default frequency causes ramp-time errors to be reported in `reports/tCW001105_1_2lsi_wc.violations`.

Maximum Clock Frequency

The maximum frequency is 195.16 MHz for lsi_wc10.

Maximum frequency heavily depends on the system. The estimate of maximum clock frequency is based on internal flip-flop to flip-flop timing.

Minimum and Maximum Duty Cycles

This design has been rated through static timing analysis for 30/70 and 70/30 duty cycles. The design is nearly independent of duty cycle because the design is fully synchronous other than the lock-up latches for scan.

Critical Path

The critical path is given below:

- Startpoint: `uArm9/uCORECTL/uIPIPE/IDarmDeint_reg_7_`
- Endpoint: `uArm9/uCOREDP/uREGBANK/BData0Ex_reg_20_`

Input AC Timings

Table 4 shows the worst-case AC timing values for the CW001105 inputs when presented with the loading of one G12P111HS compiled memory on each of the instruction and data buses. All timing in Table 4 is measured in nanoseconds.

In Table 4, the setup margin is 0 ps @ 195.16 MHz. The hold margin is approximately 52 ps. The setup and hold times are measured with respect to CLK.

Table 4 Input Signals

Signal	Setup (ns)	Hold (ns)	Notes
CHSDE	2.20	–0.02	
CHSEX	2.20	–0.02	
CPDIN	2.17	–0.02	to Output
DBGDEWPT	0.82	–0.04	
DBGEN	0.50	0.05	
DBGEXT	0.50	0.03	
DBGIEBKPT	0.63	–0.02	
DBGnTRST	1.31	0.06	to CD pin
DBGSDOUT	0.62	–0.19	to Output
DBGTCKEN	1.82	0.12	
DBGTDI	0.88	0.03	
DBGTMS	1.12	–0.03	
DMAA	0.50	0.02	
DMAD	0.50	0.02	
DMAENABLE	0.41	0.06	
DMAMAS	0.56	0.03	
DMAnREQ	0.57	0.00	
DMAnRW	0.50	0.03	
DMAWait	0.50	0.03	
DRDATA	1.74	–0.05	
DRDATA2	1.24	–0.13	
EDBGRQ	0.50	0.05	
ETMEN	1.55	0.14	

Table 4 Input Signals (Cont.)

Signal	Setup (ns)	Hold (ns)	Notes
FIFOFULL	0.87	–0.08	
HCLKEN	3.57	0.00	to Output
HGRANT	1.21	0.06	
HRDATA	1.92	–0.03	
HREADY	3.54	–0.05	to Output
HRESETn	1.59	–0.26	
HRESP	2.67	–0.09	to Output
INITRAM	1.85	–0.08	to Output
IRDATA	1.68	–0.03	
nFIQ	1.07	0.05	
nIRQ	1.06	0.05	
SCANEN	3.30	0.19	
SI0	1.25	–0.06	
SI1	1.30	–0.05	
SI2	1.28	–0.06	
SI3	1.30	–0.05	
TAPID	0.70	–0.03	
VINITHI	1.14	–0.05	

Output AC Timings

Table 5 shows the worst-case AC timing values for the CW001105 outputs when presented with the loading of one G12P111HS compiled memory on each of the instruction and data buses. All timing in Table 5 is measured in nanoseconds.

In Table 5, all synchronous outputs depend on the CLK input. The output load assumed for characterization is [8 * load_of(N1DFP/A)].

Note: The output pins of this CoreWare® component require buffering at the periphery of the macrocell to minimize loading effects on the CW001105's performance.

Table 5 Output Signals

Signal	Maximum Output Valid	Output Hold Time
BIGENDOUT	0.88	0.27
COMMRX	1.07	0.33
COMMTX	1.65	0.53
CPCLKEN	0.89	0.26
CPDOUT	0.74	0.17
CPINSTR	0.71	0.14
CPLATECANCEL	0.57	0.17
CPPASS	0.56	0.17
CPTBIT	0.49	0.15
DADDR	3.70	0.20
DADDR2	0.82	0.17
DBGACK	2.23	0.30
DBGINSTREXEC	1.57	0.27
DBGIR	0.84	0.13
DBGnTDOEN	2.01	0.34
DBGRNG	2.23	0.30
DBGRQI	1.06	0.24
DBGSCREG	0.79	0.23
DBGSDIN	0.65	0.20
DBGTAPSM	0.81	0.14
DBGTDO	2.34	0.19
DENABLE	4.16	0.45
DENABLE2	1.38	0.29
DMARData	0.55	0.13

Table 5 Output Signals (Cont.)

Signal	Maximum Output Valid	Output Hold Time
DMAReady	1.37	0.45
DWDATA	2.92	0.24
DWDATA2	0.80	0.16
DWE	3.25	0.26
DWE2	1.70	0.33
ETMBIGEND	0.48	0.14
ETMCHSD	0.50	0.12
ETMCHSE	0.53	0.13
ETMDA	0.71	0.12
ETMDABORT	0.51	0.14
ETMDBGACK	0.50	0.13
ETMDMAS	0.50	0.12
ETMDMORE	0.50	0.12
ETMDnMREQ	0.50	0.13
ETMDnRW	0.50	0.13
ETMDSEQ	0.50	0.13
ETMHIVECS	0.51	0.13
ETMIA	0.68	0.12
ETMID15To11	0.68	0.13
ETMID31To25	0.65	0.12
ETMInMREQ	0.49	0.13
ETMINSTREXEC	0.50	0.13
ETMINSTRVALID	0.50	0.13
ETMISEQ	0.65	0.16
ETMITBIT	0.51	0.13
ETMLATECANCEL	0.50	0.13
ETMnWAIT	0.52	0.16

Table 5 Output Signals (Cont.)

Signal	Maximum Output Valid	Output Hold Time
ETMPASS	0.50	0.13
ETMPROCID	0.51	0.12
ETMPROCIDWR	0.50	0.13
ETMRDATA	0.75	0.12
ETMRNGOUT	0.65	0.12
ETMWDATA	0.69	0.12
HADDR	0.87	0.18
HBURST	0.80	0.17
HBUSREQ	0.70	0.17
HLOCK	0.83	0.23
HPROT	0.80	0.21
HSIZE	0.90	0.21
HTRANS	0.97	0.23
HWDATA	0.86	0.21
HWRITE	0.86	0.23
IADDR	3.52	0.21
IENABLE	4.10	0.33
IWDATA	2.78	0.37
IWE	3.10	0.19
nCPMREQ	0.68	0.19
nCPTRANS	0.40	0.12
SO0	0.65	0.13
SO1	0.65	0.13
SO2	0.65	0.13
SO3	0.75	0.19

Clock Tree Description

The clock tree begins with a single LCLKBUF1250FP. The output of this buffer subsequently drives the clock buffer tree. There are 5 levels in the clock tree.

Clock Skew and Delays

Table 6 shows the clock timings.

Table 6 Clock Timings

Parameter	Value	
Clock Skew	approximately 108 ps (lsi_wc10)	
Insertion Delay	1.06 – 1.17 ns (lsi_wc10)	0.48 – 0.54 ns (lsi_bc)
Minimum Clock Delay	uArm9/uCOREDP/uREGBANK/ uREGBANKMEM/Reg20_reg_12_/CP (lsi_wc10)	uArm9/uCOREDP/uREGBANK/ uREGBANKMEM/Reg0_reg_13_/CP (lsi_bc)
Maximum Clock Delay	uETMIF/ETMRDATA_reg_15_/CP (lsi_wc10)	uArm9/uDbg/uDbgdp/ uICEdp/uWptdp1/DMask_reg_13_/CP (lsi_bc)

Guidelines

This section provides guidelines for the subsystem integration of the clock and STA.

Subsystem Integration of Clock Guidelines

The system must match the insertion delay of the CoreWare component.

STA Guidelines

In the scripts provided for the static timing analysis of the CW001105, all the necessary clock definitions, clock periods, false paths, multicycle paths, zero-cycle paths, and constants are defined for the developer. Refer to the script for specifics.

As a qualitative analysis of the design for STA, the developer must be aware of the following:

- No generated or gated clocks are in the design.
- There are no false paths in the critical path for this design.
- No multicycle paths are identified for this design.
- No zero-cycle paths are identified for this design.
- SCANEN is tied LOW for normal operation. It is recommended that hold times be checked with SCANEN tied HIGH as well as LOW.

Headquarters

LSI Logic Corporation
North American Headquarters
Milpitas CA
Tel: 408.433.8000
Fax: 408.433.8989

LSI Logic Europe Ltd
European Headquarters
Bracknell England
Tel: 44.1344.426544
Fax: 44.1344.481039

LSI Logic K.K.
Headquarters
Tokyo Japan
Tel: 81.3.5463.7821
Fax: 81.3.5463.7820

To receive product literature, visit us at http://www.lsilogic.com.
For a current list of our distributors, sales offices, and design resource centers, view our web page located at http://www.lsilogic.com/contacts/na_salesoffices.html.

ISO 9000 Certified

EH

Doc. No. DB08-000172-00