Welcome to E-XFL.COM

What is "Embedded - Microcontrollers"?

"Embedded - Microcontrollers" refer to small, integrated circuits designed to perform specific tasks within larger systems. These microcontrollers are essentially compact computers on a single chip, containing a processor core, memory, and programmable input/output peripherals. They are called "embedded" because they are embedded within electronic devices to control various functions, rather than serving as standalone computers. Microcontrollers are crucial in modern electronics, providing the intelligence and control needed for a wide range of applications.

Applications of "Embedded - Microcontrollers"

Details	
Product Status	Obsolete
Core Processor	ARM9®
Core Size	16/32-Bit
Speed	96MHz
Connectivity	CANbus, EBI/EMI, I²C, Microwire, SPI, SSP, UART/USART
Peripherals	Brown-out Detect/Reset, DMA, Motor Control PWM, POR, PWM, WDT
Number of I/O	80
Program Memory Size	256KB (256K x 8)
Program Memory Type	FLASH
EEPROM Size	-
RAM Size	64K x 8
Voltage - Supply (Vcc/Vdd)	1.65V ~ 2V
Data Converters	A/D 8x10b
Oscillator Type	Internal
Operating Temperature	-40°C ~ 85°C (TA)
Mounting Type	Surface Mount
Package / Case	128-LQFP
Supplier Device Package	-
Purchase URL	https://www.e-xfl.com/product-detail/stmicroelectronics/str910fw32x6

2.4.2 Branch Cache (BC)

When instruction addresses are not sequential, such as a program branch situation, the PFQ would have to flush and reload which would cause the CPU to stall if no BC were present. Before reloading, the PFQ checks the BC to see if it contains the desired target branch address. The BC contains up to four of the most recently taken branch addresses and the first four instructions associated with each of these branches. This check is extremely fast, checking all four BC entries simultaneously for a branch address match (cache hit). If there is a hit, the BC rapidly supplies the instruction and reduces the CPU stall. This gives the PFQ time to start pre-fetching again while the CPU consumes these four instructions from the BC. The advantage here is that program loops (very common with embedded control applications) run very fast if the address of the loops are contained in the BC.

In addition, there is a 5th branch cache entry that is dedicated to the Vectored Interrupt Controller (VIC) to further reduce interrupt latency by eliminating the stall latency typically imposed by fetching the instruction that reads the interrupt vector address from the VIC.

2.4.3 Management of literals

Typical ARM architecture and compilers do not place literals (data constants) sequentially in Flash memory with the instructions that use them, but instead the literals are placed at some other address which looks like a program branch from the PFQ's point of view. The STR91xF implementation of the ARM966E-S core has special circuitry to prevent flushing the PFQ when literals are encountered in program flow to keep performance at a maximum.

2.5 SRAM (64K or 96K Bytes)

A 32-bit wide SRAM resides on the CPU's Data TCM (D-TCM) interface, providing single-cycle data accesses. As shown in *Figure 1*, the D-TCM shares SRAM access with the Advanced High-performance Bus (AHB). Sharing is controlled by simple arbitration logic to allow the DMA unit on the AHB to also access to the SRAM.

2.5.1 Arbitration

Zero-wait state access occurs for either the D-TCM or the AHB when only one of the two is requesting SRAM. When both request SRAM simultaneously, access is granted on an interleaved basis so neither requestor is starved, granting one 32-bit word transfer to each requestor before relinquishing SRAM to the other. When neither the D-TCM or the AHB are requesting SRAM, the arbiter leaves access granted to the most recent user (if D-TCM was last to use SRAM then the D-TCM will not have to arbitrate to get access next time).

The CPU may execute code from SRAM through the AHB. There are no wait states as long as the D-TCM is not contending for SRAM access and the AHB is not sharing bandwidth with peripheral traffic. The ARM966E-S CPU core has a small pre-fetch queue built into this instruction path through the AHB to look ahead and fetch instructions during idle bus cycles.

2.5.2 Battery backup

When a battery is connected to the designated battery backup pin (VBATT), SRAM contents are automatically preserved when the normal operating voltage on VDD pins is lost or sags below threshold. Automatic switchover to SRAM can be disabled by firmware if it is desired that the battery will power only the RTC and not the SRAM during standby.

2.6 DMA data movement

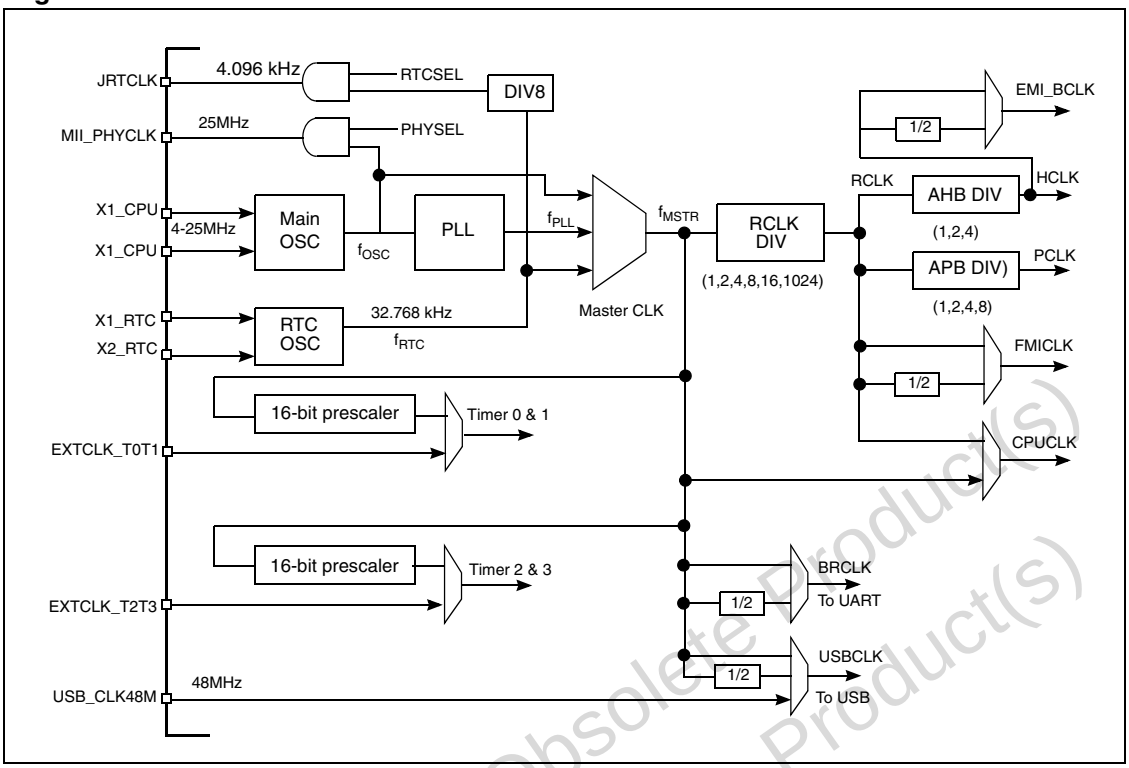
DMA channels on the Advanced High-performance Bus (AHB) take full advantage of the separate data path provided by the Harvard architecture, moving data rapidly and largely independent of the instruction path. There are two DMA units, one is dedicated to move data between the Ethernet interface and SRAM, the other DMA unit has eight programmable channels with 16 request signals to service other peripherals and interfaces (USB, SSP, I2C, UART, Timers, EMI, and external request pins). Both single word and burst DMA transfers are supported. Memory-to-memory transfers are supported in addition to memory-peripheral transfers. DMA access to SRAM is shared with D-TCM accesses, and arbitration is described in *Section 2.5.1*. Efficient DMA transfers are managed by firmware using linked list descriptor tables. Of the 16 DMA request signals, two are assigned to external inputs. The DMA unit can move data between external devices and resources inside the STR91xF through the EMI bus.

2.7 Non-volatile memories

There are two independent 32-bit wide Burst Flash memories enabling true read-while-write operation. The Flash memories are single-voltage erase/program with 20 year minimum data retention and 100K minimum erase cycles. The primary Flash memory is much larger than the secondary Flash.

Both Flash memories are blank when devices are shipped from ST. The CPU can boot only from Flash memory (configurable selection of which Flash bank).

Figure 2. Clock control



2.10.2 Reference clock (RCLK)

The main clock (f_{MSTR}) can be divided to operate at a slower frequency reference clock (RCLK) for the ARM core and all the peripherals. The RCLK provides the divided clock for the ARM core, and feeds the dividers for the AHB, APB, External Memory Interface, and FMI units.

2.10.3 AHB clock (HCLK)

The RCLK can be divided by 1, 2 or 4 to generate the AHB clock. The AHB clock is the bus clock for the AHB bus and all bus transfers are synchronized to this clock. The maximum HCLK frequency is 96 MHz.

2.10.4 APB clock (PCLK)

The RCLK can be divided by 1, 2, 4 or 8 to generate the APB clock. The APB clock is the bus clock for the APB bus and all bus transfers are synchronized to this clock. Many of the peripherals that are connected to the AHB bus also use the PCLK as the source for external bus data transfers. The maximum PCLK frequency is 48 MHz.

2.10.5 Flash memory interface clock (FMICLK)

The FMICLK clock is an internal clock derived from RCLK, defaulting to RCLK frequency at power up. The clock can be optionally divided by 2. The FMICLK determines the bus bandwidth between the ARM core and the Flash memory. Typically, codes in the Flash memory can be fetched one word per FMICLK clock in burst mode. The maximum FMICLK frequency is 96MHz.

Debugging requires that an external host computer, running debug software, is connected to the STR91xF target system via hardware which converts the stream of debug data and commands from the host system's protocol (USB, Ethernet, etc.) to the JTAG EmbeddedICE-RT protocol on the STR91xF. These protocol converters are commercially available and operate with debugging software tools.

The CPU may be forced into a Debug State by a breakpoint (code fetch), a watchpoint (data access), or an external debug request over the JTAG channel, at which time the CPU core and memory system are effectively stopped and isolated from the rest of the system. This is known as Halt Mode and allows the internal state of the CPU core, memory, and peripherals to be examined and manipulated. Typical debug functions are supported such as run, halt, and single-step. The EmbeddedICE-RT logic supports two hardware compare units. Each can be configured to be either a watchpoint or a breakpoint. Breakpoints can also be data-dependent.

Debugging (with some limitations) may also occur through the JTAG interface while the CPU is running full speed, known as Monitor Mode. In this case, a breakpoint or watchpoint will not force a Debug State and halt the CPU, but instead will cause an exception which can be tracked by the external host computer running monitor software. Data can be sent and received over the JTAG channel without affecting normal instruction execution. Time critical code, such as Interrupt Service Routines may be debugged real-time using Monitor Mode.

2.15.4 JTAG security bit

This is a non-volatile bit (Flash memory based), which when set will not allow the JTAG debugger or JTAG programmer to read the Flash memory contents.

Using JTAG ISP, this bit is typically programmed during manufacture of the end product to prevent unwanted future access to firmware intellectual property. The JTAG Security Bit can be cleared only by a JTAG "Full Chip Erase" command, making the STR91xF device blank and ready for programming again. The CPU can read the status of the JTAG Security Bit, but it may not change the bit value.

2.16 Embedded trace module (ARM ETM9, v. r2p2)

The ETM9 interface provides greater visibility of instruction and data flow happening inside the CPU core by streaming compressed data at a very high rate from the STR91xF though a small number of ETM9 pins to an external Trace Port Analyzer (TPA) device. The TPA is connected to a host computer using USB, Ethernet, or other high-speed channel. Real-time instruction flow and data activity can be recorded and later formatted and displayed on the host computer running debugger software, and this software is typically integrated with the debug software used for EmbeddedICE-RT functions such as single-step, breakpoints, etc. Tracing may be triggered and filtered by many sources, such as instruction address comparators, data watchpoints, context ID comparators, and counters. State sequencing of up to three triggers is also provided. TPA hardware is commercially available and operates with debugging software tools.

The ETM9 interface is nine pins total, four of which are data lines, and all pins can be used for GPIO after tracing is no longer needed. The ETM9 interface is used in conjunction with the JTAG interface for trace configuration. When tracing begins, the ETM9 engine compresses the data by various means before broadcasting data at high speed to the TPA over the four data lines. The most common ETM9 compression technique is to only output address information when the CPU branches to a location that cannot be inferred from the source code. This means

- Supports USB low and full-speed transfers (12 Mbps), certified to comply with the USB 2.0 specification
- Supports isochronous, bulk, control, and interrupt endpoints
- Configurable number of endpoints allowing a mixture of up to 20 single-buffered monodirectional endpoints or up to 10 double-buffered bidirectional endpoints
- Dedicated, dual-port 2 Kbyte USB Packet Buffer SRAM. One port of the SRAM is connected by a Packet Buffer Interface (PBI) on the USB side, and the CPU connects to the other SRAM port.
- CRC generation and checking
- NRZI encoding-decoding and bit stuffing
- USB suspend resume operations

2.18.1 Packet buffer interface (PBI)

The PBI manages a set of buffers inside the 2 Kbyte Packet Buffer, both for transmission and reception. The PBI will choose the proper buffer according to requests coming from the USB Serial Interface Engine (SIE) and locate it in the Packet SRAM according to addresses pointed by endpoint registers. The PBI will also auto-increment the address after each exchanged byte until the end of packet, keeping track of the number of exchanged bytes and preventing buffer overrun. Special support is provided by the PBI for isochronous and bulk transfers, implementing double-buffer usage which ensures there is always an available buffer for a USB packet while the CPU uses a different buffer.

2.18.2 DMA

A programmable DMA channel may be assigned by CPU firmware to service the USB interface for fast and direct transfers between the USB bus and SRAM with little CPU involvement. This DMA channel includes the following features:

- Direct USB Packet Buffer SRAM to system SRAM transfers of receive packets, by descriptor chain for bulk or isochronous endpoints.
- Direct system SRAM to USB Packet Buffer SRAM transfers of transmit packets, by descriptor chain for bulk or isochronous endpoints.
- Linked-list descriptor chain support for multiple USB packets

2.18.3 Suspend mode

CPU firmware may place the USB interface in a low-power suspend mode when required, and the USB interface will automatically wake up asynchronously upon detecting activity on the USB pins.

2.19 CAN 2.0B interface

The STR91xF provides a CAN interface complying with CAN protocol version 2.0 parts A and B. An external CAN transceiver device connected to pins CAN_RX and CAN_TX is required for connection to the physical CAN bus.

The CAN interface manages up to 32 Message Objects and Identifier Masks using a Message SRAM and a Message Handler. The Message Handler takes care of low-level CAN bus activity such as acceptance filtering, transfer of messages between the CAN bus and the Message

2.21 I²C interfaces with DMA

The STR91xF supports two independent I2C serial interfaces, designated I2C0, and I2C1. Each interface allows direct connection to an I2C bus as either a bus master or bus slave device (firmware configurable). I2C is a two-wire communication channel, having a bi-directional data signal and a single-directional clock signal based on open-drain line drivers, requiring external pull-up resistors.

Byte-wide data is transferred between a Master device and a Slave device on two wires. More than one bus Master is allowed, but only one Master may control the bus at any given time. Data is not lost when another Master requests the use of a busy bus because I2C supports collision detection and arbitration. More than one Slave device may be present on the bus, each having a unique address. The bus Master initiates all data movement and generates the clock that permits the transfer. Once a transfer is initiated by the Master, any device that is addressed is considered a Slave. Automatic clock synchronization allows I2C devices with different bit rates to communicate on the same physical bus. A single device can play the role of Master or Slave, or a single device can be a Slave only. A Master or Slave device has the ability to suspend data transfers if the device needs more time to transmit or receive data.

Each I2C interface on the STR91xF has the following features:

- Programmable clock supports various rates up to I2C Standard rate (100 KHz) or Fast rate (400 KHz).
- Serial I/O Engine (SIOE) takes care of serial/parallel conversion; bus arbitration; clock generation and synchronization; and handshaking
- Multi-master capability
- 7-bit or 10-bit addressing

2.21.1 DMA

A programmable DMA channel may be assigned by CPU firmware to service each I2C channel for fast and direct transfers between the I2C bus and SRAM with little CPU involvement. Both DMA single-transfers and DMA burst-transfers are supported for transmit and receive.

2.22 SSP interfaces (SPI, SSI, and Microwire) with DMA

The STR91xF supports two independent Synchronous Serial Port (SSP) interfaces, designated SSP0, and SSP1. Primary use of each interface is for supporting the industry standard Serial Peripheral Interface (SPI) protocol, but also supporting the similar Synchronous Serial Interface (SSI) and Microwire communication protocols.

SPI is a three or four wire synchronous serial communication channel, capable of full-duplex operation. In three-wire configuration, there is a clock signal, and two data signals (one data signal from Master to Slave, the other from Slave to Master). In four-wire configuration, an additional Slave Select signal is output from Master and received by Slave.

The SPI clock signal is a gated clock generated from the Master and regulates the flow of data bits. The Master may transmit at a variety of baud rates, up to 24 MHz

In multi-Slave operation, no more than one Slave device can transmit data at any given time. Slave selection is accomplished when a Slave's "Slave Select" input is permanently grounded or asserted active-low by a Master device. Slave devices that are not selected do not interfere with SPI activities. Slave devices ignore the clock signals and keep their data output pins in

2.25.1 DMA

A programmable DMA channel may be assigned by CPU firmware to service each timer/ counter module TIM0 and TIM1 for fast and direct transfers.

2.26 Three-phase induction motor controller (IMC)

The STR91xF provides an integrated controller for variable speed motor control applications.

Six PWM outputs are generated on high current drive pins P6.0 to P6.5 for controlling a three-phase AC induction motor drive circuit assembly. Rotor speed feedback is provided by capturing a tachometer input signal on pin P6.6, and an asynchronous hardware emergency stop input is available on pin P6.7 to stop the motor immediately if needed, independently of firmware.

The IMC unit has the following features:

- Three PWM outputs generated using a 10-bit PWM counter, one for each phase U, V, W. Complimentary PWM outputs are also generated for each phase.
- Choice of classic or zero-centered PWM generation modes
- 10-bit PWM counter clock is supplied through a programmable 8-bit prescaler of the APB clock.
- Programmable 6-bit dead-time generator to add delay to each of the three complimentary PWM outputs
- 8-bit repetition counter
- Automatic rotor speed measurement with 16-bit resolution. Schmitt trigger tachometer input with programmable edge detection
- Hardware asynchronous emergency stop input
- A dedicated interrupt to CPU with eight flags

2.27 External memory interface (EMI)

STR91xF devices in 128-pin packages offer an external memory bus for connecting external parallel peripherals and memories. The EMI bus resides on ports 7, 8, and 9 and operates with either an 8 or 16-bit data path. The configuration of 8 or 16 bit mode is specified by CPU firmware writing to configuration registers at run-time. If the application does not use the EMI bus, then these port pins may be used for general purpose I/O as shown in *Table 3*.

The EMI has the following features:

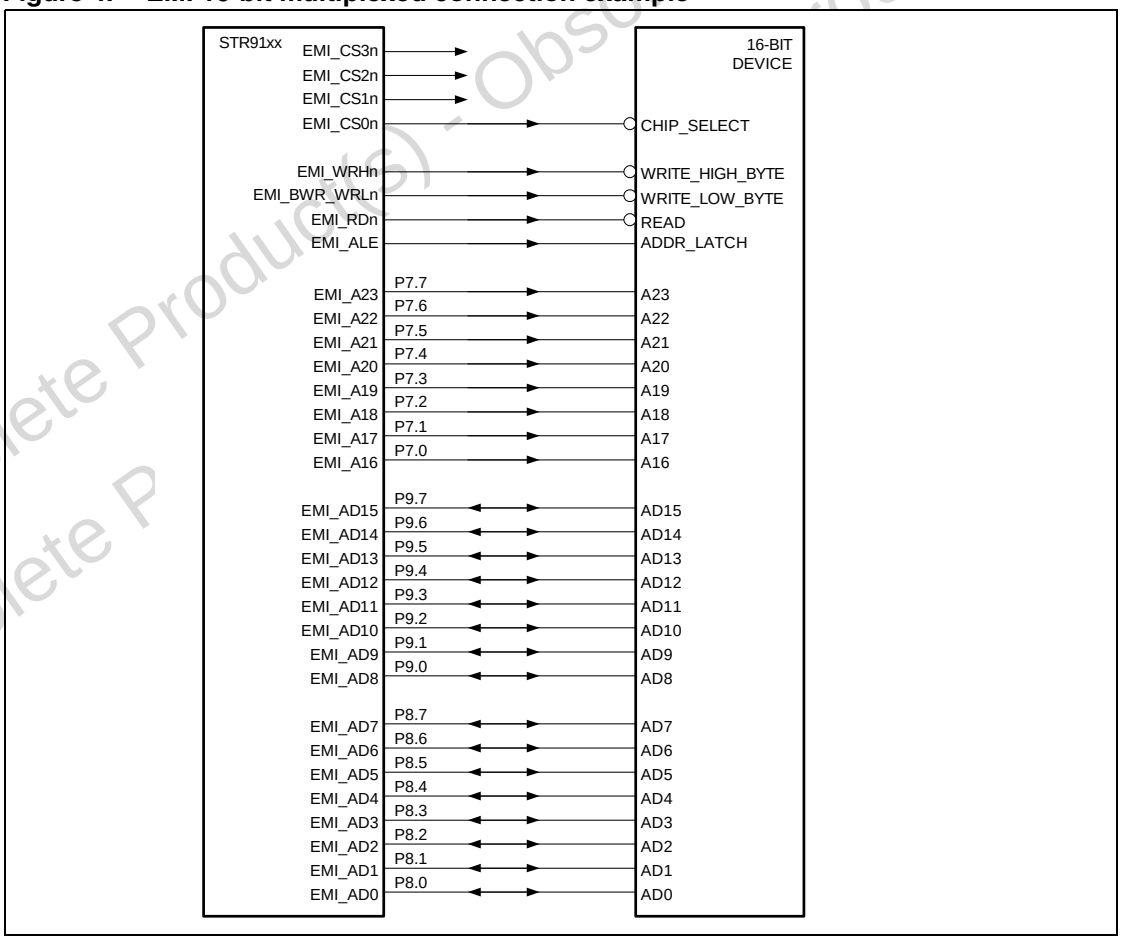
- Supports static asynchronous memory access cycles, including page mode for non-mux operation.
- Four configurable memory regions, each with a chip select output (EMI_CS0n ... EMI_CS3n)
- Programmable wait states per memory region for both write and read operations
- **16-bit multiplexed data mode** (*Figure 4*): 16 bits of data and 16 bits of low-order address are multiplexed together on ports 8 and 9, while port 7 contains eight more high-order address signals. The output signal on pin EMI_ALE is used to demultiplex the signals on ports 8 and 9, and the polarity of EMI_ALE is programmable. The output signals on pins EMI_BWR_WRLn and EMI_WRHn are the write strobes for the low and high data bytes

respectively. The output signal EMI_RDn is the read strobe for both the low and high data bytes.

- **8-bit multiplexed data mode**: This is a variant of the 16-bit multiplexed mode. Although this mode can provide 24 bits of address and 8 bits of data, it does require an external latch device on Port 8. However, this mode is most efficient when connecting devices that only require 8 bits of address on an 8-bit multiplexed address/data bus, and have simple read, write, and latch inputs as shown in *Figure 5*

  To use all 24 address bits, the following applies: 8 bits of lowest-order data and 8 bits of lowest-order address are multiplexed on port 8. On port 9, 8-bits of mid-order address are multiplexed with 8 bits of data, but these 8 data values are always at logic zero on this port during a write operation, and these 8 data bits are ignored during a read operation. An external latch device (such as a '373 latch) is needed to de-multiplex the mid-order 8 address bits that are generated on port 8. Port 7 outputs the 8 highest-order address signals directly (not multiplexed). The output signal on pin EMI_ALE is used to demultiplex the signals on ports 8 and 9, and the polarity of EMI_ALE is programmable. The output signal on pin EMI_BWR_WRLn is the data write strobe, and the output on pin EMI_RDn is the data read strobe.
- **8-bit non-multiplexed data mode** (*Figure 6*): Eight bits of data are on port 8, while 16 bits of address are output on ports 7 and 9. The output signal on pin EMI_BWR_BWLn is the data write strobe and the output on pin EMI_RDn is the data read strobe.

Figure 4. EMI 16-bit multiplexed connection example



Pkg		Pin Name	Signal Type	Default Pin Function	Default Input Function	Alternate functions			
LQFP80	LQFP128					Alternate Input 1	Alternate Output 1	Alternate Output 2	Alternate Output 3
-	5	P7.0	I/O	GPIO_7.0, GP Input, HiZ	EXINT24, External Intr	TIM0_ICAP1, Input Capture	GPIO_7.0, GP Output	8b) EMI_A0, 16b) EMI_A16	ETM_PCK0, *ETM Packet*
-	6	P7.1	I/O	GPIO_7.1, GP Input, HiZ	EXINT25, External Intr	TIM0_ICAP2, Input Capture	GPIO_7.1, GP Output	8b) EMI_A1, 16b) EMI_A17	ETM_PCK1, *ETM Packet*
-	7	P7.2	I/O	GPIO_7.2, GP Input, HiZ	EXINT26, External Intr	TIM2_ICAP1, Input Capture	GPIO_7.2, GP Output	8b) EMI_A2, 16b) EMI_A18	ETM_PCK2, *ETM Packet*
-	13	P7.3	I/O	GPIO_7.3, GP Input, HiZ	EXINT27, External Intr	TIM2_ICAP2, Input Capture	GPIO_7.3, GP Output	8b) EMI_A3, 16b) EMI_A19	ETM_PCK3, *ETM Packet*
-	14	P7.4	I/O	GPIO_7.4, GP Input, HiZ	EXINT28, External Intr	UART0_RxD, UART rcv data	GPIO_7.4, GP Output	8b) EMI_A4, 16b) EMI_A20	EMI_CS3n, EMI Chip Select
-	15	P7.5	I/O	GPIO_7.5, GP Input, HiZ	EXINT29, External Intr	ETM_EXTRIG, ETM ext. trigger	GPIO_7.5, GP Output	8b) EMI_A5, 16b) EMI_A21	EMI_CS2n, EMI Chip Select
-	118	P7.6	I/O	GPIO_7.6, GP Input, HiZ	EXINT30, External Intr	TIM3_ICAP1, Input Capture	GPIO_7.6, GP Output	8b) EMI_A6, 16b) EMI_A22	EMI_CS1n, EMI Chip Select
-	119	P7.7	I/O	GPIO_7.7, GP Input, HiZ	EXINT31, External Intr	TIM3_ICAP2, Input Capture	GPIO_7.7, GP Output	EMI_CS0n, EMI chip select	16b) EMI_A23, 8b) EMI_A7
-	26	P8.0	I/O	GPIO_8.0, GP Input, HiZ	-	-	GPIO_8.0, GP Output	8b) EMI_D0, 16b) EMI_AD0	-
-	28	P8.1	I/O	GPIO_8.1, GP Input, HiZ	-	-	GPIO_8.1, GP Output	8b) EMI_D1, 16b) EMI_AD1	-
-	30	P8.2	I/O	GPIO_8.2, GP Input, HiZ	-	-	GPIO_8.2, GP Output	8b) EMI_D2, 16b) EMI_AD2	-
-	32	P8.3	I/O	GPIO_8.3, GP Input, HiZ	-	-	GPIO_8.3, GP Output	8b) EMI_D3, 16b) EMI_AD3	-
-	34	P8.4	I/O	GPIO_8.4, GP Input, HiZ	-	-	GPIO_8.4, GP Output	8b) EMI_D4, 16b) EMI_AD4	-
-	36	P8.5	I/O	GPIO_8.5, GP Input, HiZ	-	-	GPIO_8.5, GP Output	8b) EMI_D5, 16b) EMI_AD5	-
-	38	P8.6	I/O	GPIO_8.6, GP Input, HiZ	-	-	GPIO_8.6, GP Output	8b) EMI_D6, 16b) EMI_AD6	-
-	44	P8.7	I/O	GPIO_8.7, GP Input, HiZ	-	-	GPIO_8.7, GP Output	8b) EMI_D7, 16b) EMI_AD7	-
-	46	P9.0	I/O	GPIO_9.0, GP Input, HiZ	-	-	GPIO_9.0, GP Output	8b) EMI_A8 16b) EMI_AD8	-
-	47	P9.1	I/O	GPIO_9.1, GP Input, HiZ	-	-	GPIO_9.1, GP Output	8b) EMI_A9, 16b) EMI_AD9	-
-	50	P9.2	I/O	GPIO_9.2, GP Input, HiZ	-	-	GPIO_9.2, GP Output	8b) EMI_A10, 16b)EMI_AD10	-
-	51	P9.3	I/O	GPIO_9.3, GP Input, HiZ	-	-	GPIO_9.3, GP Output	8b) EMI_A11, 16b)EMI_AD11	-
-	52	P9.4	I/O	GPIO_9.4, GP Input, HiZ	-	-	GPIO_9.4, GP Output	8b) EMI_A12, 16b)EMI_AD12	-
-	58	P9.5	I/O	GPIO_9.5, GP Input, HiZ	-	-	GPIO_9.5, GP Output	8b) EMI_A13, 16b)EMI_AD13	-
-	62	P9.6	I/O	GPIO_9.6, GP Input, HiZ	-	-	GPIO_9.6, GP Output	8b) EMI_A14, 16b)EMI_AD14	-
-	64	P9.7	I/O	GPIO_9.7, GP Input, HiZ	-	-	GPIO_9.7, GP Output	8b) EMI_A15, 16b)EMI_AD15	-

5 Memory mapping

The ARM966E-S CPU addresses a single linear address space of 4 giga-bytes (2^{32}) from address 0x0000.0000 to 0xFFFF.FFFF as shown in *Figure 9*. Upon reset the CPU boots from address 0x0000.0000, which is chip-select zero at address zero in the Flash Memory Interface (FMI).

The Instruction TCM and Data TCM enable high-speed CPU operation without incurring any performance or power penalties associated with accessing the system buses (AHB and APB). I-TCM and D-TCM address ranges are shown at the bottom of the memory map in *Figure 9*.

5.1 Buffered and non-buffered writes

The CPU makes use of write buffers on the AHB and the D-TCM to decouple the CPU from any wait states associated with a write operation. The user may choose to use write with buffers on the AHB by setting bit 3 in control register CP15 and selecting the appropriate AHB address range when writing. By default at reset, buffered writes are disabled (bit 3 of CP15 is clear) and all AHB writes are non-buffered until enabled. *Figure 9* shows that most addressable items on the AHB are aliased at two address ranges, one for buffered writes and another for non-buffered writes. A buffered write will allow the CPU to continue program execution while the write-back is performed through a FIFO to the final destination on the AHB. If the FIFO is full, the CPU is stalled until FIFO space is available. A non-buffered write will impose an immediate delay to the CPU, but results in a direct write to the final AHB destination, ensuring data coherency. Read operations from AHB locations are always direct and never buffered.

5.2 System (AHB) and peripheral (APB) buses

The CPU will access SRAM, higher-speed peripherals (USB, Ethernet, Programmable DMA), and the external bus (EMI) on the AHB at their respective base addresses indicated in *Figure 9*. Lower-speed peripherals reside on the APB and are accessed using two separate AHB-to-APB bridge units (APB0 and APB1). These bridge units are essentially address windows connecting the AHB to the APB. To access an individual APB peripheral, the CPU will place an address on the AHB bus equal to the base address of the appropriate bridge unit APB0 or APB1, plus the offset of the particular peripheral, plus the offset of the individual data location within the peripheral. *Figure 9* shows the base addresses of bridge units APB0 and APB1, and also the base address of each APB peripheral. Please consult the STR91xF Reference manual for the address of data locations within each individual peripheral.

5.3 SRAM

The SRAM is aliased at three separate address ranges as shown in *Figure 9*. When the CPU accesses SRAM starting at 0x0400.0000, the SRAM appears on the D-TCM. When CPU access starts at 0x4000.0000, SRAM appears in the buffered AHB range. Beginning at CPU address 0x5000.0000, SRAM is in non-buffered AHB range. The SRAM size must be specified by CPU intitialization firmware writing to a control register after any reset condition. Default SRAM size is 32K bytes, with option to set to 64K bytes on STR91xFx32 devices, and to 96K bytes on STR91xFx44 devices.

When other AHB bus masters (such as a DMA controller) write to SRAM, their access is never buffered. Only the CPU can make use of buffered AHB writes.

5.4 Two independent Flash memories

The STR91xF has two independent Flash memories, the larger primary Flash and the small secondary Flash. It is possible for the CPU to erase/write to one of these Flash memories while simultaneously reading from the other.

One or the other of these two Flash memories may reside at the “boot” address position of 0x0000.0000 at power-up or at reset as shown in *Figure 9*. The default configuration is that the first sector of primary Flash memory is enabled and residing at the boot position, and the secondary Flash memory is disabled. This default condition may be optionally changed as described below.

5.4.1 Default configuration

When the primary Flash resides at boot position, typical CPU initialization firmware would set the start address and size of the main Flash memory, and go on to enable the secondary Flash, define it’s start address and size. Most commonly, firmware would place the secondary Flash start address at the location just after the end of the primary Flash memory. In this case, the primary Flash is used for code storage, and the smaller secondary flash can be used for data storage (EEPROM emulation).

5.4.2 Optional configuration

Using the STR91xF device configuration software tool, or IDE from 3rd party, one can specify that the smaller secondary Flash memory is at the boot location at reset and the primary Flash is disabled. The selection of which Flash memory is at the boot location is programmed in a non-volatile Flash-based configuration bit during JTAG ISP. The boot selection choice will remain as the default until the bit is erased and re-written by the JTAG interface. The CPU cannot change this choice for boot Flash, only the JTAG interface has access.

In this case where the secondary Flash defaults to the boot location upon reset, CPU firmware would typically initialize the Flash memories the following way. The secondary Flash start address and size is specified, then the primary Flash is enabled and its start address and size is specified. The primary Flash start address would typically be located just after the final address location of the secondary Flash. This configuration is particularly well-suited for In-Application-Programming (IAP). The CPU would boot from the secondary Flash memory, initialize the system, then check the contents of the primary Flash memory (by checksum or other means). If the contents of primary Flash is OK, then CPU execution continues from either Flash memory. If the main Flash contents are incorrect, the CPU, while executing code from the secondary Flash, can download new data from any STR91xF communication channel and program into primary Flash memory. Application code then starts after the new contents of primary Flash are verified.

6 Electrical characteristics

6.1 Absolute maximum ratings

This product contains devices to protect the inputs against damage due to high static voltages. However, it is advisable to take normal precautions to avoid application of any voltage higher than the specified maximum rated voltages. It is also recommended to ground any unused input pin to reduce power consumption and minimize noise.

Table 4. Absolute maximum ratings

Symbol	Parameter	Value		Unit
		Min	Max	
V_{DD}	Voltage on VDD pin with respect to ground V_{SS}	-0.3	2.4	V
V_{DDQ}	Voltage on VDDQ pin with respect to ground V_{SS}	-0.3	4.0	V
V_{BATT}	Voltage on VBATT pin with respect to ground V_{SS}	-0.3	4.0	V
AV_{DD}	Voltage on AVDD pin with respect to ground V_{SS} (128-pinpackage)	-0.3	4.0	V
AV_{REF}	Voltage on AVREF pin with respect to ground V_{SS} (128-pin package)	-0.3	4.0	V
AV_{REF_AVDD}	Voltage on AVREF_AVDD pin with respect to Ground V_{SS} (80-pin package)	-0.3	4.0	V
V_{IN}	Voltage on 5V tolerant pins with respect to ground V_{SS}	-0.3	5.5	V
	Voltage on any other pin with respect to ground V_{SS}	-0.3	4.0	V
I_{OV}	Input current on any pin during overload condition	-10	+10	mA
I_{TDV}	Absolute sum of all input currents during overload condition		|200|	mA
T_{ST}	Storage Temperature	-55	+150	°C
T_J	Junction Temperature		+125	°C
ESD	ESD Susceptibility (Human Body Model)	2000		V

Note: Stresses exceeding above listed recommended "Absolute Maximum Ratings" may cause permanent damage to the device. This is a stress rating only and functional operation of the device at these or any other conditions above those indicated in the operational sections of this specification is not implied. Exposure to absolute maximum rating conditions for extended periods may affect device reliability. During overload conditions ($V_{IN}>V_{DDQ}$ or $V_{IN}<V_{SSQ}$) the voltage on pins with respect to ground (V_{SSQ}) must not exceed the recommended values.

6.4 DC electrical characteristics

V_{DDQ} = 2.7 - 3.6V, V_{DD} = 1.65 - 2V, T_A = -40 / 85 °C unless otherwise specified.

Table 7. DC Electrical Characteristics

Symbol	Parameter	Test Conditions	Value			Unit
			Min	Typ	Max	
V_{IH}	Input High Level	General inputs	2.0		(1)	V
		RESET and TCK inputs	$0.8V_{DDQ}$			
V_{IL}	Input Low Level	General inputs			0.8	
		RESET and TCK inputs			$0.2V_{DDQ}$	
V_{HYS}	Input Hysteresis Schmitt Trigger	General inputs	0.4			V
V_{OH}	Output High Level High current pins	I/O ports 3 and 6: Push-Pull, I_{OH} = 8mA	V_{DDQ}-0.7			V
	Output High Level Standard current pins	I/O ports 0,1,2,4,5,7,8,9: Push-Pull, I_{OH} = 4mA	V_{DDQ}-0.7			
V_{OL}	Output Low Level High current pins	I/O ports 3 and 6: Push-Pull, I_{OL} = 8mA			0.4	V
	Output Low Level Standard current pins	I/O ports 0,1,2,4,5,7,8,9: Push-Pull, I_{OL} = 4mA			0.4	

Notes: 1 Input pins are 5V tolerant, max input voltage is 5.5V

6.5 AC electrical characteristics

V_{DDQ} = 2.7 - 3.6V, V_{DD} = 1.65 - 2V, T_A = -40 / 85 °C unless otherwise specified.

Table 8. AC electrical characteristics

Symbol	Parameter	Test Conditions		Value			Unit
				Min	Typ	Max	
I_{DDRUN}	Run Mode Current	All peripherals on	CPU_CLK = 96MHz [1] [5]		1.7	2.3	mA/ MHz
		All peripherals off			1.3	1.6	
I_{IDLE}	Idle Mode Current	All peripherals on [2] [5]			1.14	1.7	mA/ MHz
		All peripherals off [3] [5]			0.45	0.75	mA/ MHz
I_{SLEEP}	Sleep Mode Current	LVD On [4] [5]			55	825	µA
		LVD Off [4] [5]			50	820	µA
I_{RTC_STBY}	RTC Standby Current	Measured on VBATT pin			0.3	0.9	µA
I_{SRAM_STBY}	SRAM Standby Current	Measured on VBATT pin			5	85	µA

Notes: 1 ARM core and peripherals active with all clocks on. Power can be conserved by turning off clocks to peripherals which are not required.

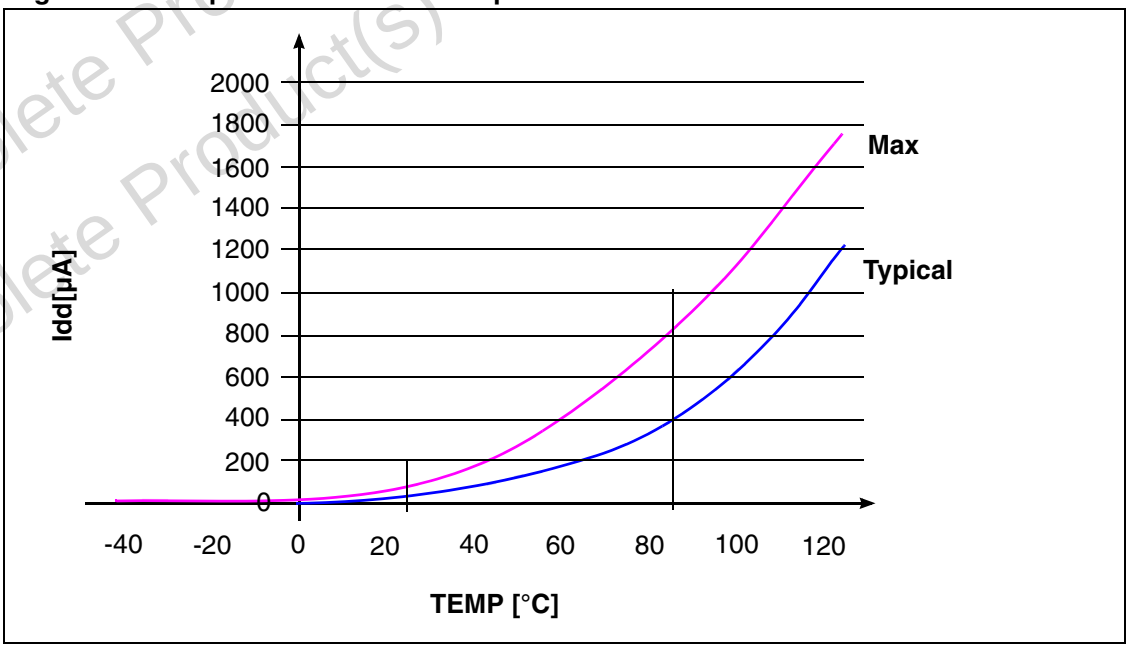
2 ARM core stopped and all peripheral clocks active.

3 ARM core stopped and all peripheral clocks stopped.

4 ARM core and all peripheral clocks stopped (with exception of RTC).

5 Current measured on the V_{DD} pins. V_{DDQ} current is not included.

Figure 10. Sleep Mode current vs temperature



6.7 Main oscillator electrical characteristics

V_{DDQ} = 2.7 - 3.6V, V_{DD} = 1.65 - 2V, T_A = -40 / 85 °C unless otherwise specified.

Table 11. Main oscillator electrical characteristics

Symbol	Parameter	Test Conditions	Value			Unit
			Min	Typ	Max	
$t_{STUP(OSC)}$	Oscillator Start-up Time	Stable V_{DDQ}			3	mS

6.8 RTC oscillator electrical characteristics

V_{DDQ} = 2.7 - 3.6V, V_{DD} = 1.65 - 2V, T_A = -40 / 85 °C unless otherwise specified.

Table 12. RTC oscillator electrical characteristics

Symbol	Parameter	Test Conditions	Value			Unit
			Min	Typ	Max	
$g_{M(RTC)}$	Oscillator Start _voltage		LVD [1)]			V
$t_{STUP(RTC)}$	Oscillator Start-up Time	Stable V_{DDQ}			1	S

Notes: 1 Min oscillator start voltage is the same as low voltage detect level (2.4V or 2.7V) for VDDQ

Table 13. RTC crystal electrical characteristics

Symbol	Parameter	Test Conditions	Value			Unit
			Min	Typ	Max	
f_O	Resonant frequency			32.768		kHz
R_S	Series resistance				40	kΩ
C_L	Load capacitance			8		pF

Symbol	Parameter	Value		Unit
		Min	Max	
t_{AAH}	Address to ALE hold time	$t_{BCLK/2}$ - 1	$t_{BCLK/2}$ + 1	ns
t_{AAS}	Address to ALE setup time	(ALE_LENGTH) x t_{BCLK} - 1		ns

Notes: 1 ALE_LENGTH = 1 by default (can be programmed to be 2 by setting the bits In the SCU_SCR0 register)

2 WSTRD = 1Fh by default (RD wait state time = WSTRD x t_{BCLK}, WSTRD can be programmed in the EMI_RCRx Register)

3 WSTOEN = 1 by default (RD assertion delay from chip select. WSTOEN can be programmed in the EMI_OECRx Register)

Figure 11. Non-mux bus (8-bit) read timings

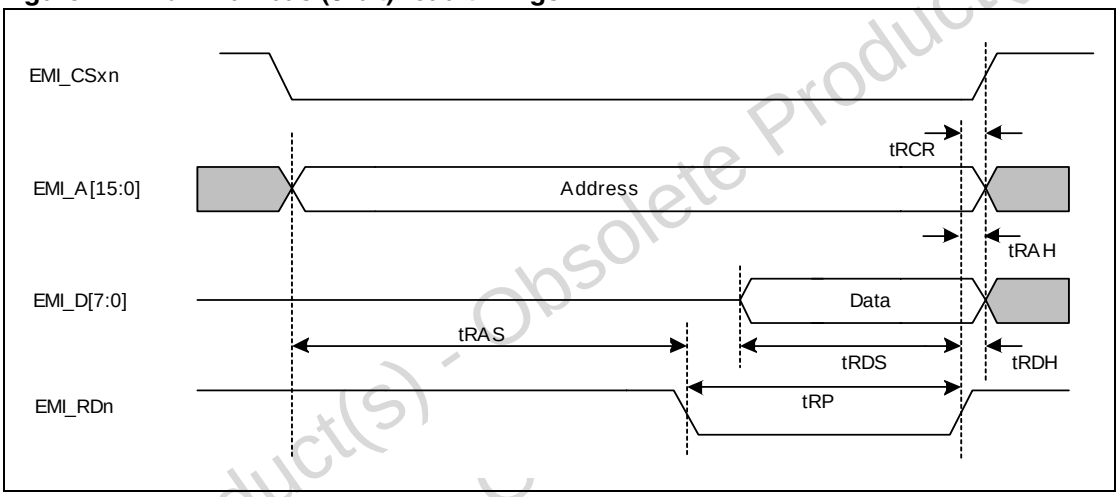
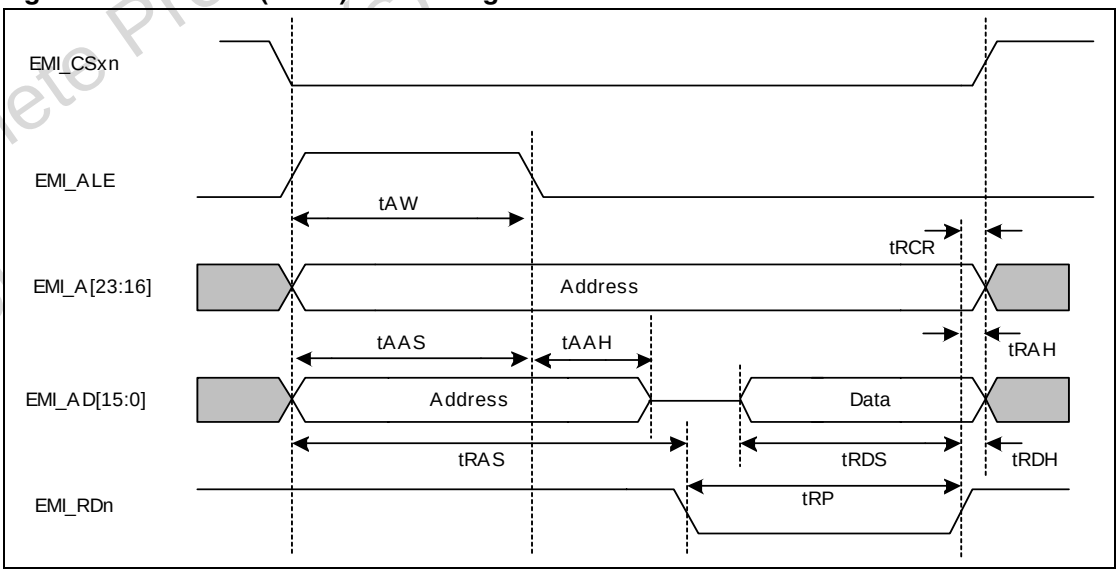


Figure 12. Mux bus (16-bit) read timings



6.14.4 I^2C electrical characteristics

V_{DDQ} = 2.7 - 3.6V, V_{DD} = 1.65 - 2V, T_A = -40 / 85 °C unless otherwise specified.

Table 28. I^2C Electrical Characteristics

Symbol	Parameter	Standard I^2C		Fast I^2C		Unit
		Min	Max	Min	Max	
t_{BUF}	Bus free time between a STOP and START condition	4.7		1.3		ms
$t_{HD:STA}$	Hold time START condition. After this period, the first clock pulse is generated	4.0		0.6		µs
t_{LOW}	LOW period of the SCL clock	4.7		1.3		µs
t_{HIGH}	HIGH period of the SCL clock	4.0		0.6		µs
$t_{SU:STA}$	Set-up time for a repeated START condition	4.7		0.6		µs
$t_{HD:DAT}$	Data hold time	0		0		ns
$t_{SU:DAT}$	Data set-up time	250		100		ns
t_R	Rise time of both SDA and SCL signals		1000	$20+0.1C_b$	300	ns
t_F	Fall time of both SDA and SCL signals		300	$20+0.1C_b$	300	ns
$t_{SU:STO}$	Set-up time for STOP condition	4.0		0.6		µs
C_b	Capacitive load for each bus line		400		400	pF

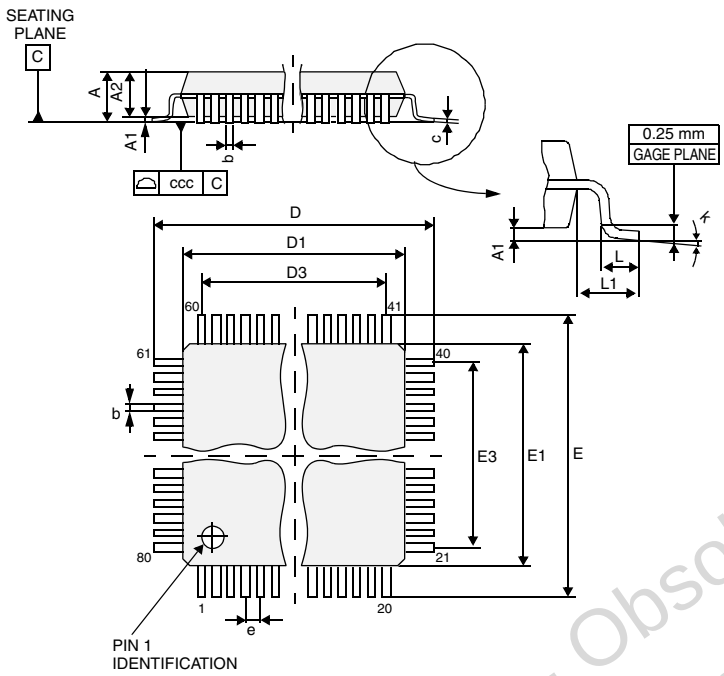
Notes: 1 The device must internally provide a hold time of at least 300 ns for the SDA signal in order to bridge the undefined region of the falling edge of SCL

2 The maximum hold time of the START condition has only to be met if the interface does not stretch the low period of SCL signal

3 C_b = total capacitance of one bus line in pF

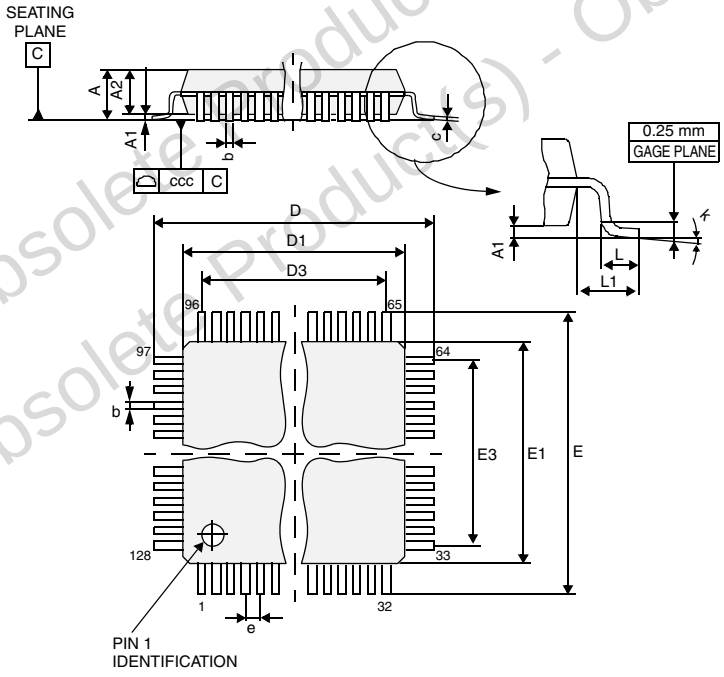
7 Package mechanical data

Figure 25. 80-Pin Low Profile Quad Flat Package



Dim.	mm			inches		
	Min	Typ	Max	Min	Typ	Max
A			1.60			0.063
A1	0.05		0.15	0.002		0.006
A2	1.35	1.40	1.45	0.053	0.055	0.057
b	0.17	0.22	0.27	0.007	0.009	0.011
c	0.09		0.20	0.004		0.008
D		14.00			0.551	
D1		12.00			0.472	
D2		9.50			0.374	
E		14.00			0.551	
E1		12.00			0.472	
E2		9.50			0.374	
e		0.50			0.020	
L	0.45	0.60	0.75	0.018	0.024	0.030
L1		1.00			0.039	
k	0d		7d	0d		7d
ddd			0.08			0.003
	Number of Pins					
N	80					

128-Pin Low Profile Quad Flat Package



Dim.	mm			inches		
	Min	Typ	Max	Min	Typ	Max
A			1.60			0.063
A1	0.05		0.15	0.002		0.006
A2	1.35	1.40	1.45	0.053	0.055	0.057
b	0.13	0.18	0.23	0.005	0.007	0.009
c	0.09		0.20	0.004		0.008
D	15.80	16.00	16.20	0.622	0.630	0.638
D1	13.80	14.00	14.20	0.543	0.551	0.559
D3		12.40			0.488	
E	15.80	16.00	16.20	0.622	0.630	0.638
E1	13.80	14.00	14.20	0.543	0.551	0.559
E3		12.40			0.488	
e		0.40			0.016	
L	0.45	0.60	0.75	0.018	0.024	0.030
L1		1.00			0.039	
k	0d	3.5d	7d	0d	3.5d	7d
ccc			0.08			0.003
	Number of Pins					
N	128					

8 Ordering information

Table 31. Ordering information

Part Number	Flash KB	RAM KB	Major Peripherals	Package[1)]
STR910FM32X6	256+32	64	CAN, 40 I/Os	LQFP80, 12x12 mm
STR910FW32X6	256+32	64	CAN, EMI, 80 I/Os	LQFP128, 14x14 mm
STR911FM42X6	256+32	96	USB, CAN, 40 I/Os	LQFP80, 12x12mm
STR911FM44X6	512+32	96		
STR912FW42X6	256+32	96	Ethernet, USB, CAN, EMI, 80 I/Os	LQFP128
STR912FW44X6	512+32	96		

Table 32. Ordering information scheme

Example: STR9 1 2 F W 4 4 X 6 T

Family

ARM9 Microcontroller Family

Series

1 = STR9 Series 1

Feature set

0 = CAN, UART, IrDA, I2C, SSP

1 = USB, CAN, UART, IrDA, I2C, SSP

2 = USB, CAN, UART, IrDA, I2C, SSP, ETHERNET

Memory type

F = Flash

No. of pins

M = 80

W = 128

SRAM size

3 = 64K

4 = 96K

Program Memory Size

2 = 256K

4 = 512K

Package

X = plastic LQFP

Temperature Range

6 = -40 to 85°C

Shipping Option

T = Tape & Reel Packing

For a list of available options (e.g. speed, package) or for further information on any aspect of this device, please contact the ST Sales Office nearest to you.