ST22N256

Smartcard 32-Bit RISC MCU with 256 Kbytes EEPROM Javacard™ HW Execution & Cryptographic Library

DATA BRIEF

PRODUCT FEATURES

- 32-BIT RISC CPU WITH 24-BIT LINEAR MEMORY ADDRESSING
- 368 KBYTES USER ROM
- 16 KBYTES USER RAM
- 256K KBYTES USER EEPROM

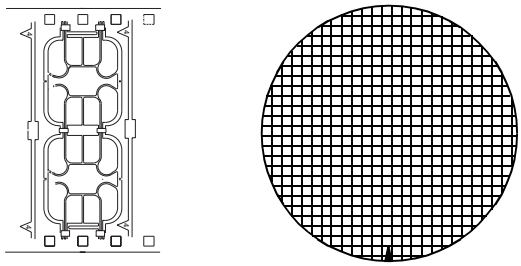
32-BIT RISC CPU

- DUAL INSTRUCTION SET, JAVACARD™ AND NATIVE
- 4-STAGE PIPELINE
- 16 GENERAL PURPOSE 32-BIT REGISTERS, AND SPECIAL REGISTERS
- 4 MASKABLE INTERRUPT LEVELS
- SUPERVISOR AND USER MODES

SECURITY

- CPU SECURITY INSTRUCTIONS
  - DES and 3DES instructions
  - Fast Multiply and Accumulate instructions for Public Key and Elliptic Curve Cryptography
  - CRC instruction
- RANDOM NUMBER GENERATOR
- EEPROM FLASH PROGRAMMING MODE
- CLOCK AND POWER MANAGEMENT
- VOLTAGE AND CLOCK FREQUENCY SENSORS
- ADVANCED MEMORY PROTECTION
  - *Memory Protection Unit* for application firewalling and peripheral access control
  - Domain switching securely controlled by protected Context Stack
  - Native/Java, Code/Data memory attributes with 128-byte granularity
- FOUR WORKING STACKS
  - Java stack with both 16 and 32-bit accesses
  - User and Supervisor mode stacks
  - Security Context stack

Figure 1. Delivery Form



CRYPTOGRAPHIC LIBRARY

- ASYMMETRICAL ALGORITHMS
  - Software Crypto libraries in separate ROM area for efficient algorithm coding using a set of advanced functions. RSA, signature/ verification.
  - RSA key calculation including Prime number generation SHA-1
- SYMMETRICAL ALGORITHMS
  - DES, Triple DES, AES

CRYPTOGRAPHY PERFORMANCE

The following table provides the cryptographic performances of the ST22N256 based on ST Crypto Library.

Table 1. Preliminary Cryptographic Performances

Algorithm	Function	Time(1)
RSA 1024 bits	Signature with CRT	79.0 ms
	Signature without CRT(2)	242.0 ms
	Verification (e=0x10001)	3.6 ms
RSA 2048 bits	Signature with CRT	485.0 ms
	Signature without CRT	1.7 s
	Verification (e=0x10001)	11.0 ms
DES	Triple	18 µs
	Single	8 µs
SHA-1	512-bit Block	194 µs
AES-128	Encryption including subkey computation	85 µs

1. Internal clock at 33 MHz
2. CRT: Chinese Reminder Theorem

MEMORY

- HIGHLY RELIABLE CMOS EEPROM TECHNOLOGY
  - Error Correction Code for single bit fail within a 32-bit word
  - 10 years data retention, 500,000 Erase/Write cycles endurance
  - 1 to 128 bytes Erase or Program in 2 ms typical
- HIGH PERFORMANCE MEMORY
  - Dual memory buses for data and instruction
  - Byte, Short (2) and Word (4) load and store
  - Address auto-increment

OTHER FEATURES

- HARDWARE ASYNCHRONOUS SERIAL INTERFACE (ASI)
  - 1M baud rate capability
  - 2 serial I/O ports compatible ISO 7816-3 T=0 and T=1
- 2 USER CONFIGURABLE 12-BIT AND 16-BIT TIMERS WITH INTERRUPT
- CENTRAL INTERRUPT CONTROLLER WITH UP TO 16 INPUT LINES
- EXTERNAL CLOCK FROM 1 MHz TO 10 MHz
- 1.62 V TO 5.5 V SUPPLY VOLTAGE
- TEMPERATURE RANGE -25° C to +85° C
- POWER SAVING STANDBY MODE
- ESD PROTECTION GREATER THAN 5000 V
- UNIQUE IDENTIFICATION PER DIE
- TYPICAL INTERNAL FREQUENCY UP TO 33 MHz
- SOFTWARE CONTROLLED CLOCK MANAGEMENT

DESCRIPTION

The ST22N256 is a member of the SmartJ™ platform using a 32-bit Reduced Instruction Set Computer (RISC) core to execute both Native RISC instructions and JavaCard™ 2.x Technology instruction (byte codes) directly.

Direct JavaCard™ byte code execution provides high performance advantage over processors that emulate the JavaCard™ byte code instruction set.

The product features a 24-bit wide linear addressing capability and includes User ROM, User RAM, and User EEPROM.

Memory and Peripheral accesses are controlled by a *Memory Protection Unit* that allows to implement firewalls between applications.

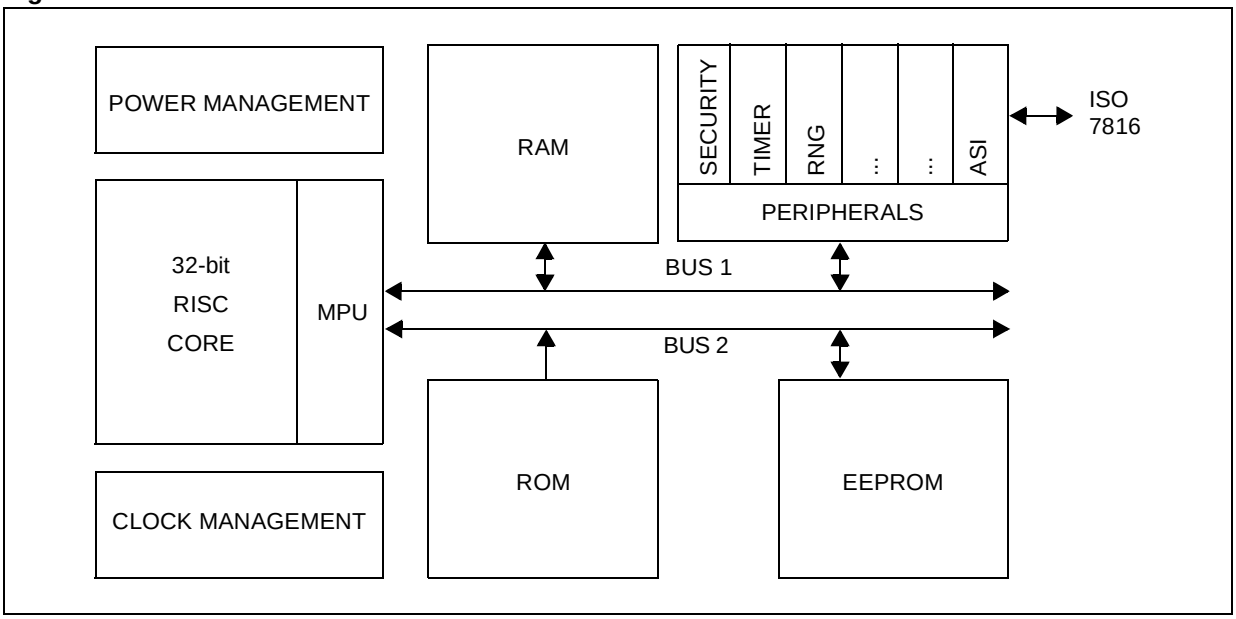
Memories are accessed via two different buses, allowing simultaneous accesses to code and data.

Memory load and stores can be performed at byte, short (2-bytes), or word (4-bytes) granularity, with optional pointer auto increment.

The ST22 core includes dedicated DES instructions for Secret Key cryptography, and a fast Multiply and Accumulate instruction for Public Key cryptography (RSA) and Elliptic Curve cryptography, and a CRC instruction. The ST22 core also includes specific instructions for security.

The product has clock and power management, 2 User configurable Timers, a Central Interrupt Controller and a Random Number Generator.

Figure 2. SmartJ™ Platform EEPROM Architecture



The product has two execution modes. *Java* mode is used when JavaCard™ 2.x byte codes are being executed. *Native* mode is used for long JavaCard™ byte codes, Native methods and system routines. The processor enters Java mode when a dispatch (*DISP*) instruction is encountered. When executing in Native mode, there are two privilege levels, *User* and *Supervisor*. Some instructions can only be executed in *Supervisor* mode.

Instructions are of variable length, from 1 to 4 bytes in Native mode.

Special instructions exist for single-cycle stack operations, a frequent occurrence in Java code. Short branches and conditional branches within a 1 KByte block or the entire 16-MByte instruction space are supported. The product has four stages of pipeline in Native mode: fetch, decode, execute and write-back. In Java mode, there are five stages of pipeline: byte code-fetch, byte code-decode, decode, execute and write-back.

The CPU core has 16 32-bit general purpose registers, as well as special registers of variable length.

The chip also features a very high performance Asynchronous Serial Interface (ASI) to support high speed serial communication protocols compatible with ISO 7816 standard.

It is manufactured using the highly reliable ST CMOS EEPROM technology.

EMBEDDED SOFTWARE

The Hardware Software Interface (HSI) implements the Hardware abstraction layer. It consists of C interfaces to the EEPROM memory and peripherals. The drivers are:

– Non Volatile Memory
– Asynchronous Serial Interface
– Central Interrupt Controller
– Timer
– Random Number Generator
– Clock Manager
– Memory Protection Unit
– Sensors
– Security

Note:

– The HSI driver software layer is a C-oriented API allowing efficient and secureaccess to the peripherals and Non Volatile Memory for programming or erasing.
– Only the OS and JavaCard™ Virtual Machine (JVM) domains can access the HSI software layer (In the following the term OS will refer to the software layer that is directly interfaced to the HSI).

CRYPTOGRAPHIC LIBRARY

ST proposes a complete set of firmware subroutines. This library is located in a specific ROM area. It saves the operating system designer from coding first layer functions and allows him to concentrate on algorithms, Public Key Cryptography and Secret Key Cryptography protocols implementation.

The cryptographic library, located in a specific ROM area, contains firmware functions for:

■ ASYMMETRICAL ALGORITHMS:
  – basic mathematics including modular squaring and multiplication for various lengths;
  – modular exponentionation;
  – more elaborate functions such as RSA signatures and verifications for modulo length up to 2048 bits long;
  – full internal key generation for signatures/ verifications. This guarantees that the secret key will never be known outside the chip and contributes to the overall system security.
  – long random number generation
  – SHA-1
  – RSA key generation
■ SYMMETRICAL ALGORITHMS
  – DES, Triple DES
  – AES-128, AES-192, AES-256

SOFTWARE DEVELOPMENT ENVIRONMENT

Modularity, flexibility and methodology are the key words for the SmartJ™ Development Tools Platform. Using the same interface, the developers are able to create, compile and debug a project.

The SmartJ™ Integrated Development environment (IDE) includes:

– A code Generation chain: C/C++ compiler, assembler and linker. The assembler supports both native and JavaCard™ instruction sets.

– An instruction set simulator, a cycle accurate simulator, a C/C++ source level debugger and hardware emulation tools.

OEM DEVELOPMENT LICENSE TYPES

The ST22N256 is a product based on the SmartJ™ Platform. Developers have two types of licenses for access to the technology:

■ **STLDA**

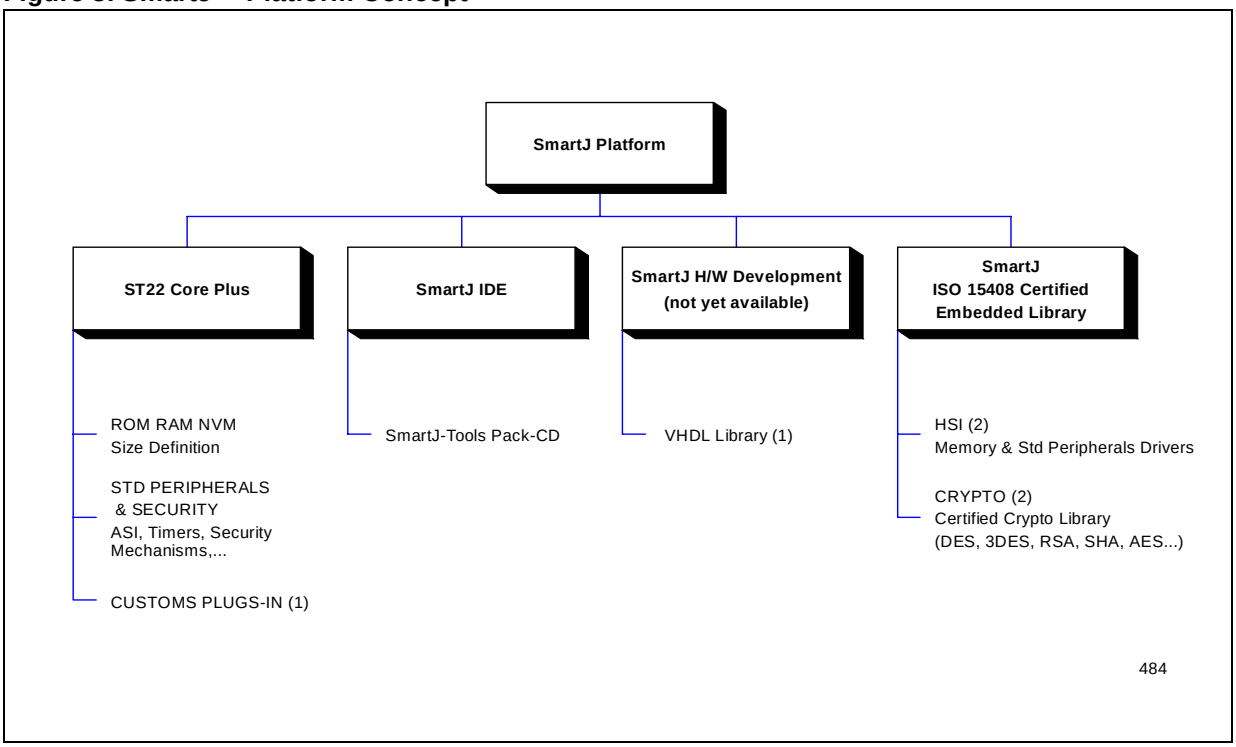
The SmartJ™ Technology License and Distribution Agreement for Standard OEM Developers (Embedded Operating System and Application Software developers) and Card Embedders.

They must use the SmartJ™ Hardware Software Interface (HSI) meta-layer communication interface to access the product hardware resources. The validation of the Embedded Software will be done using the Simulators of the Code Validation Tools chain.

■ **SPTLA (not yet available)**

The SmartJ™ Platform Technology License Agreement for OEM Platform Developers. The SPTLA is for developers who need to develop a customised architecture using the platform blocks assembled with a proprietary custom hardware plug-in logic block and associated firmware. The complete Code Validation Tools chain including the VHDL Emulator, must be used for both the hardware, software development integration and validation. The complete Code Validation Tool chain is accessible to OEM Platform Developers licensees only.

Figure 3. SmartJ™ Platform Concept



Note: **1)** SmartJ™ Platform Technology License Agreement required
2) SmartJ™ Technology License and Distribution Agreement required

Figure 4. SmartJ™ IDE

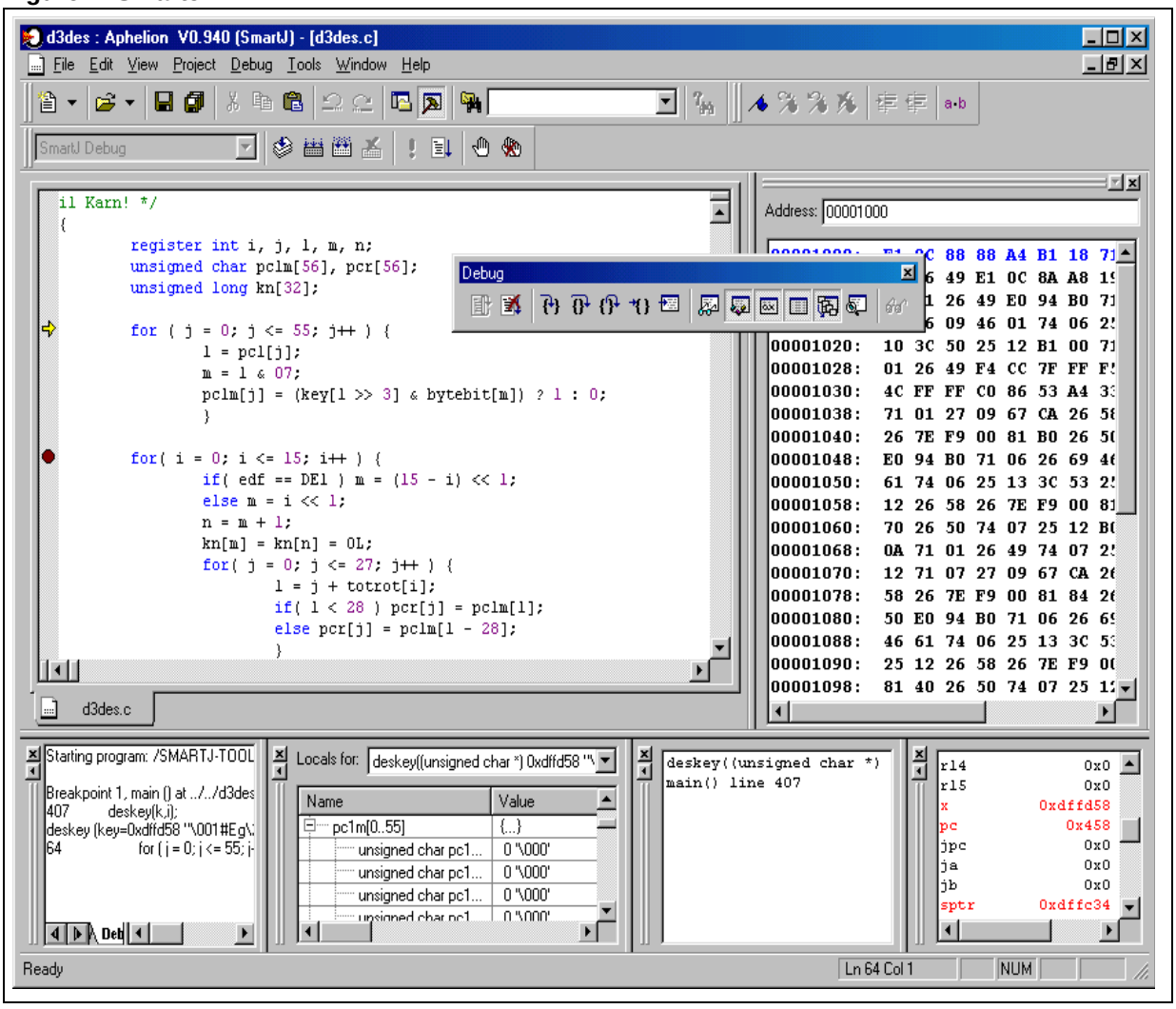


Figure 5. SmartJ™ Code Generation Tools

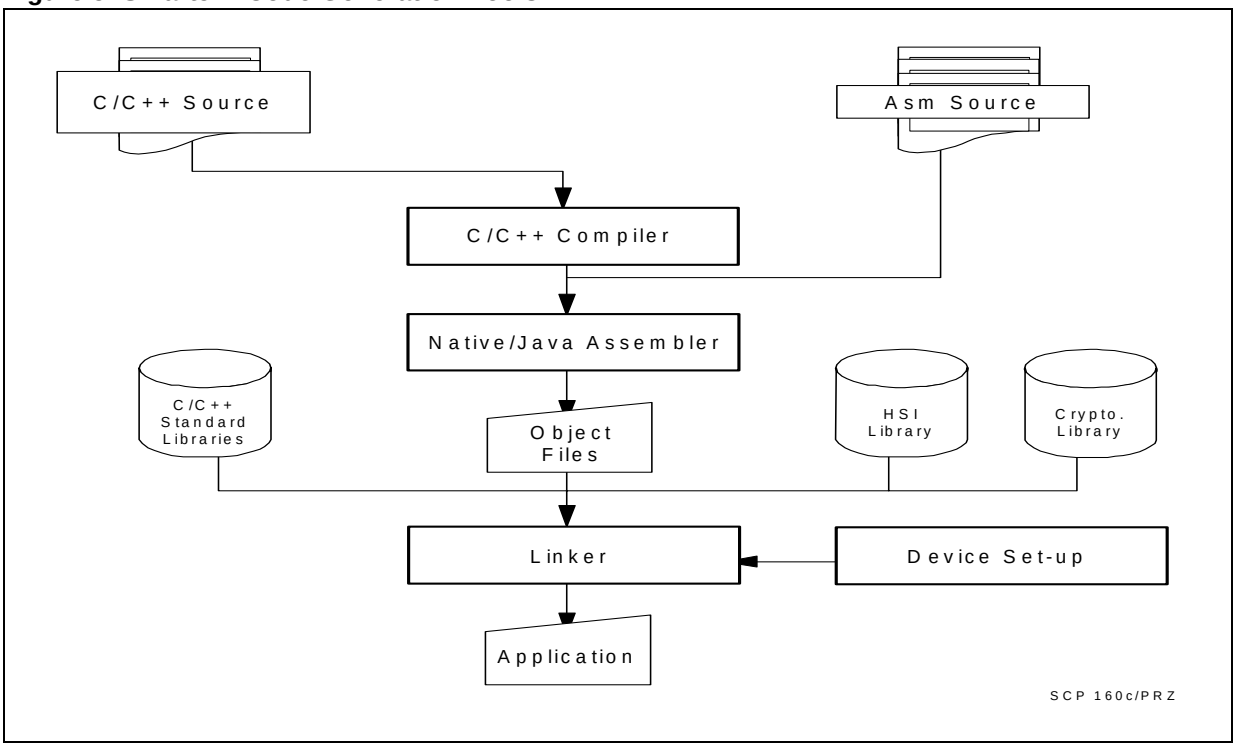


Figure 6. SmartJ™ Code Validation Tools