VINCI Chip – High-Speed Data Encryption / Decryption Unit

Description

The V_{INCI} Chip is a programmable data encryption/decryption unit designed to encrypt and decrypt 64-bit blocks of data using the International Data Encryption Algorithm (I_{DEA}™) by Lai and Massey [1, 2]. The V_{INCI} Chip operates on 64-bit data blocks using a set of encryption subkeys which are internally generated from a 128-bit user-specified key, to produce 64-bit cipher words. The operation is reversible : provided that decryption is programmed and the cipher string is input, the original plaintext string is produced. The completed ciphering process is performed on-chip ; no intermediate results are stored off-chip. However, the 128-bit key is user-defined and may be changed.

Features

- 177 Mbit/sec Data Conversion Rate (@ 25 Mhz Clock Frequency)
- Implements Lai's and Massey's Proposal for a New International Data Encryption Standard
- Standard Encrypt and Encrypt Modes ECB, CBC, CFB, OFB, and MAC Available
- Fast Encrypt and Decrypt Modes ECB8, CBC8, CFB8, and OFB8 Available
- Complies Fully to Security Standards

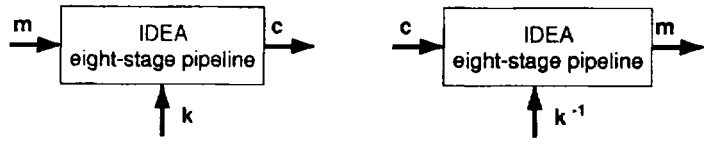
Functional Description

Introduction

The I_{DEA}™ algorithm is a *block cipher*, i.e., it is applied to encrypt or decrypt 64-bit data blocks. The cipher contains a structure (called *round*) which is replicated eight times to achieve the desired statistical properties of the cipher texte. As the cipher has the property of *similarity*[1], an output transformation step is necessary after the eight computational rounds.

The core of the V_{INCI} Chip is the encryption / decryption unit shown in Fig. 1. It contains of an eight-stage pipelined implementation of one round. The required output transform can be included in an additional round while bypassins some arithmetic operations.

Figure 1. Encrypter (left) and decrypter (right)



1. The property of similarity results in identical hardware for the encrypter and the decrypter, respectively.

I_{DEA} is a trademark of ASCOM and a joint development of ASCOM/ETHZ.

The implications of a eight-stage pipelined conversion[2] unit with nine rounds (in which the output transform is included) working is included) working on 64-bit data blocks are as follows :

Latency : Consider a completely filled pipelined : At starting time, eight data blocks must be available which are read in eight clock cycles. These eight data blocks are the converted during nine rounds. The first data block will therefore be available after t_{Lat} = 72 . t_{Cycle}, where t_{Cycle} denotes the clock cycle time. t_{Lat} is called *latency time* or just *latency*.

In case of a single block conversion, only one data block is needed at the beginning of the conversion. The latency, however, is the same as fro the case of a completely filled pipeline.

Throughput : Since one data block has a word length of 64 bits, the throughput is :

$$T = \frac{8 \cdot 64 \text{ bit}}{8 \cdot 9 \cdot t_{Cycle}} = 7.11 \text{ bit} \cdot f_{Clock}$$

where f_{Clock} denotes the clock frequency.

For single block conversion the number of pipeline stages does not influence the throughput. It is decreased by the factor eight, resulting in $T = 0.889 \text{ bit} \cdot f_{Clock}$.

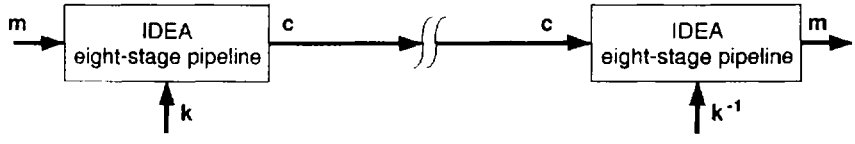
Fig. 1 shows the block diagram of the conversion unit which is called I_{DEA} *eight-stage pipeline*. Since encryption and decryption are basically the same procedure depending only on the key applied to the conversion hardware, the two blocks are identical. **m** stands for *plaintext* (the *message* to encrypt), **c** for *ciphertext* (encrypted data), **k** for the encryption key, and $\mathbf{k}^{-1}$ for the decryption key.

Fundamentals on modes of operation

The V_{INCI} Chip implements the following five standard modes of operations [5] :

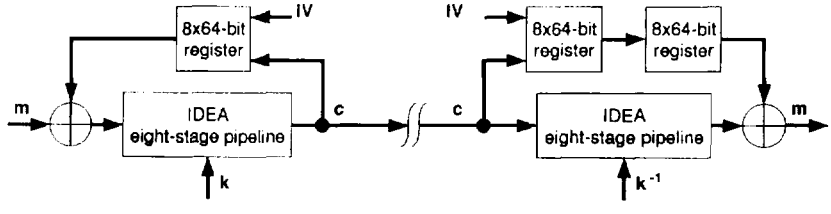
1. **ECB** : Electronic Code Book mode (Fig. 2). in this mode conversion is straightforward since there is no data feedback.

Figure 2. ECB Mode



2. **CBC** : Cipher Block Chaining mode (Fig. 3). For encryption, converted data is fed back and combined with the next cleartext block by bitwise XOR operation. An initialization vector **IV** is required to be combined with the first block to encrypt.

Figure 3. CBC Mode



For decryption the ciphertext is fed forward and combined with the next decrypted data block by bitwise XOR operation. The initialization vector **IV** is required again for correct handling of the first block to decrypt.

Data buffers are required for data synchronization. An 8 × 64-bit register is necessary for encryption, whereas two such registers are needed for decryption.

2. *Conversion* means encryption or decryption.

3. **CFB** : Cipher Feedback mode (Fig. 4). The cipher acts as pseudo-random generator with initial value **IV** which depends on the ciphertext **c**. Note that for decryption the encryption key is required.
4. **OFB** : Output Feedback mode (Fig. 5). The cipher acts as pseudo-random generator which depends only on the initial value **IV**. Note again that for decryption the encryption key is required.
5. **MAC** : Message Authentication Code. This mode is identical to CBC encryption mode, where only the last s blocks of the whole encryption are of interest [6].

Figure 4. CFB Mode

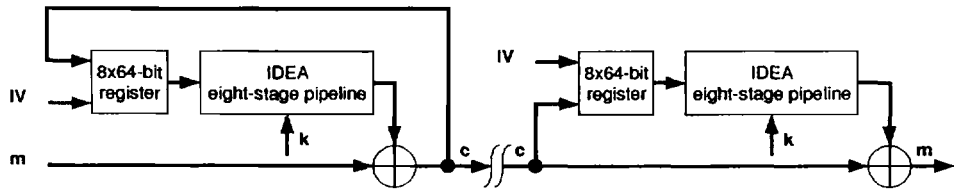
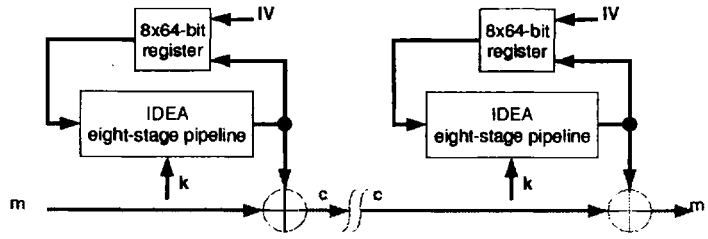


Figure 5. OFB Mode



All modes with internal feedback/feedforward structure, i.e., CBC, CFB, OFB, and MAC, cannot exploit the pipeline, since a converted block is combined with the immediately following. Consider for example the CBC mode depicted in Fig. 6. The first block to convert will not be combined with the second until it passed nine times through the eight-stage pipeline. The throughput therefore is being reduced to $T = \frac{64}{9 \cdot 8}$ bit $\cdot f_{Clock} = 0.889 \cdot f_{Clock}$.

To face this decrease in processing speed, a new scheme is proposed and has been implemented in the V_{INCI} Chip. Consider Fig. 6 again. If the first block will be combined with the ninth, the second with the tenth, the l-th $(l + 8)$-th, maximum throughput can be achieved. Compared to the standard scheme, however, eight different and independent "chains" are generated as depicted in Fig. 7. The new mode is called *CBC8*. Consequently, the standard scheme is called CBC1.

Figure 6. Encryption in CBC mode

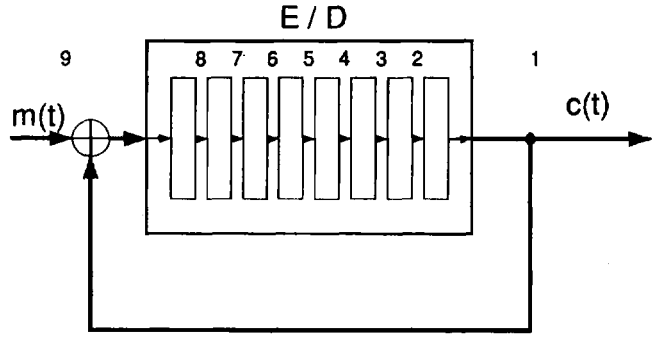
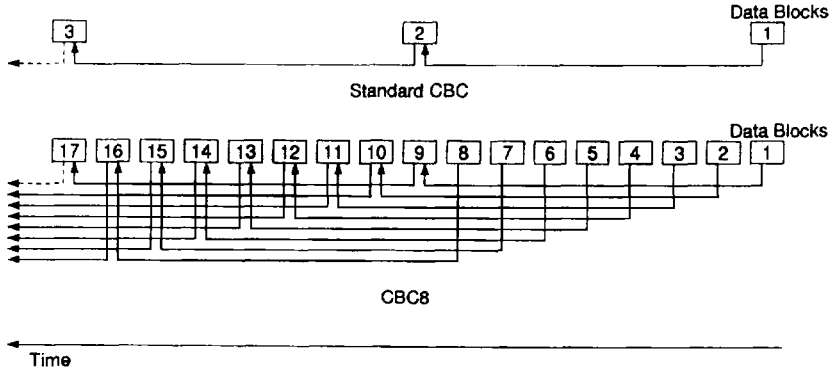


Figure 7. Dependencies in the proposed feedback/feedforward modes



Implemented modes of operations

The V_{INCI} Chip supports the following modes of operations :

- Electronic Code Book mode : ECB8 encryption, ECB8 decryption, ECB1 encryption and ECB1 decryption.
- Cipher Block Chaining mode : CBC8 encryption, CBC8 decryption, CBC1 encryption and CBC1 decryption.
- Cipher Feedback mode : CFB8 encryption, CFB8 decryption, CFB1 encryption and CFB1 decryption.
- Output Feedback mode : OFB8 encryption / decryption and OFB1 encryption/decryption. In contrast to the classical OFB scheme this mode is used as "key stream generator" only, i.e., the combination with the plaintext and ciphertext (the grey colored part in Fig. 5), respectively, has not been realized on-chip.
- Message Authentication Code : MAC1, which is identical to CBC1 encryption mode, where only the last *s* blocks of the whole encryption are of interest.

Initialization vectors

CBC, CFB, OFB, and MAC modes need an initialization until the real conversion procedure can start. The loading scheme for the initialization vectors for CBC8, CFB8, and OFB8 are as follows :

The first eight data blocks read in via the data port are regarded as initialization vectors **IV1** (first data block) until **IV8** (last data block).

The loading scheme for the initialization vectors to implement for CBC1, CFB1, and MAC are as follows :

The first data block read in via the data port is regarded as initialization vector **IV1.**

Initialization vectors can be transferred as ciphertext, i.e., they have to be decrypted before use. This has to be done in ECB8 and ECB1, respectively.

Initialization vectors are stored internally in the **IV** register.

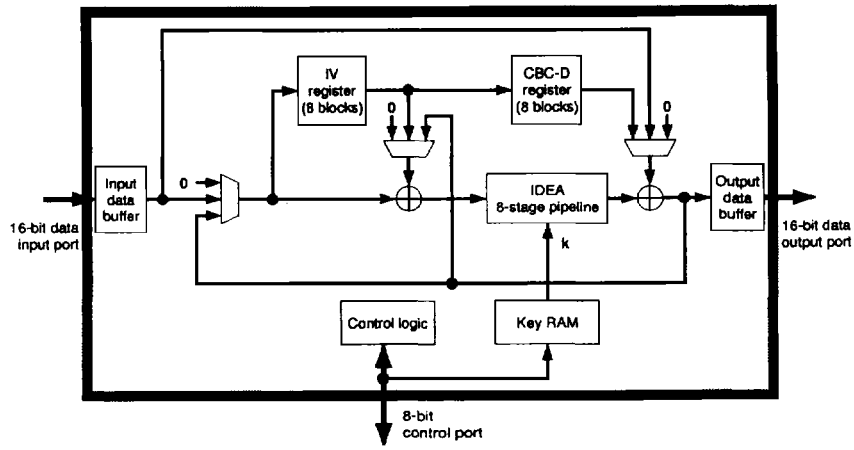
Chip Architecture And Signal Description

Overview

Fig. 8 shows the overall architecture of the V_{INCI} Chip. The core of the chip is an 8-stage pipeline with additional feedback/feedforward structures and temporary storage registers. This units implements all operational modes of conversion defined in section 1.3.

3

Figure 8. Overall architecture of the V_{INCI} Chip.



Due to the properties of the I_{DEA} algorithm the V_{INCI} Chip achieves a very high throughput rate (>177 Mbit/s in ECB8, e. g.). In view of high-speed applications the interfaces have been implemented as unidirectional data ports.

The interfaces of the V_{INCI} Chip are designed in such a way that the data can be loaded continuously and independently of the internal chip clock. Depending on the input data flow, the pipeline unit is running or remains in an idle state until the input data buffers are filled appropriated with data blocks to convert. The average throughput rate of the pipeline unit is always greater or equal in the maximum than the external data rate. It is leaved to the user to fix the chip clock frequency to the maximum value to minimize the latency of the data to convert.

The control logic is responsible for the data transfer between buffers and conversion unit and the supply of correct subkeys to the conversion unit. In addition, it does the whole key management like loading new keys or generating subkeys. The communication to the outer world is done via the control port. All parameters which define a specific conversion can be set and the actual status watched by this port.

General signals

Signal Name	Type	Active	Description
~RESET	I	L	Master reset. If this signal is tied low, all data registers[1] will be cleared[2] and the control port registers will be initialized. The control logic will be set to its initial state. After master reset, an off-line self-test is initiated. Active low schmitt trigger input.
CLOCK	I	↑	Master clock : Chip clock with positive edge active.
VDD			Power supply, + 5 V.
VSS			Ground, 0 V.

1. "Data registers" mean input and output data buffer and all pipeline registers (**IV** and temporary registers included).
2. To "clear" means that data is not visible outside the chip any more. It shows the same effect as initializing the data registers with zero.

Interfaces

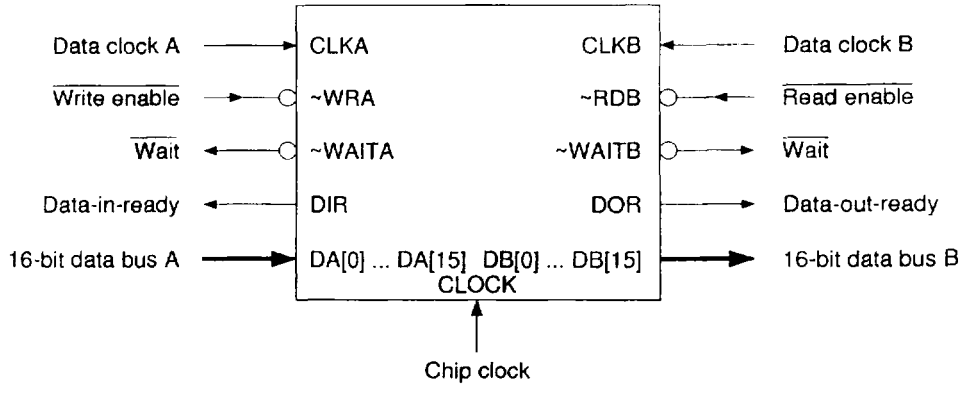
The V_{INCI} Chip is equipped with three ports, two unidirectional 16-bit data ports and an 8-bit bidirectional control port.

Data ports

The two data ports A and B are constructed in such a way that data can be loaded and read asynchronously to the chip clock CLOCK. The functionality of the buffers is equivalent to a FIFO (first-in/first-out memory) with storing capacity of eight 64-bits blocks.

Fig. 9 shows the controls signals and buses of the data ports.

Figure 9. Data Port Signals of the V_{INCI} Chip.



Data port signals

Signal Name	Type	Active	Description
CLKA	I	↑	Port A's data clock signal with positive edge active.
CLKB	I	↑	Port B's data clock signal with positive edge active.
~WRA	I	L	Port A's write enable signal. If ~WRA is low, the data on DA[0] ... DA[15] are loaded on the next rising edge of CLKA into the input buffer.
~RDB	I	L	Port B's read enable signal. If ~RDB is low, the data in the output buffer is written on DB[0] ... DB[15] on the next rising edge of CLKB.
~WAITA	O	L	Port A's wait signal is set if data has been loaded but not transferred yet to the input buffer.
~WAITB	O	L	Port B's wait signal is set when a read request is detected on port B but th next data word has not been read out yet from the output buffer.
DIR	O	H	"Data-in-ready" holds until the input buffer is filled up. DIR is set after a master reset and after the command data reset (bit 1 of the command register).
DOR	O	H	"Data-out-ready" holds until the output buffer is empty. DOR is inactive after a master reset and after the command "data reset" (bit 1 of the command register).
DA[0] ... DA[15]	I	H	Data bus A for input data, IV vectors, and session keys. LSB is DA[0].
DB[0] ... DB[15]	O	H	Tristate data bus B for output data. LSB is DB[0]

3

Interface status signals

Three dedicated signals show status information of the current conversion :

- The key multiplex signal KMUX shows whether key 1 (usually an encryption key) or key 2 (usually a decryption key) is currently in use. This signal allows to operate the V_{INCI} Chip in multiplexing mode for two data streams, where the one is converted by key 1 and the other by key 2. Key 1 and key 2 are determined by the key select register.
- The synchronization signal SYNC shows the status of the pipeline and the buffers. The edges of the signal describe the moment at which the input buffer is emptied and the output buffer is filled, respectively.
- The chip clock signal CLK2 is the half of the clock frequency supplied to **CLOCK.** Connected directly to **CLKA** and **CLKB** it can be used to transfer data on and off the chip synchronously to the chip clock.

The timing of the two signals **KMUX** and **SYNC** is shown in Fig. 10 and Fig. 11.

Figure 10. Timing of interface signals at average speed

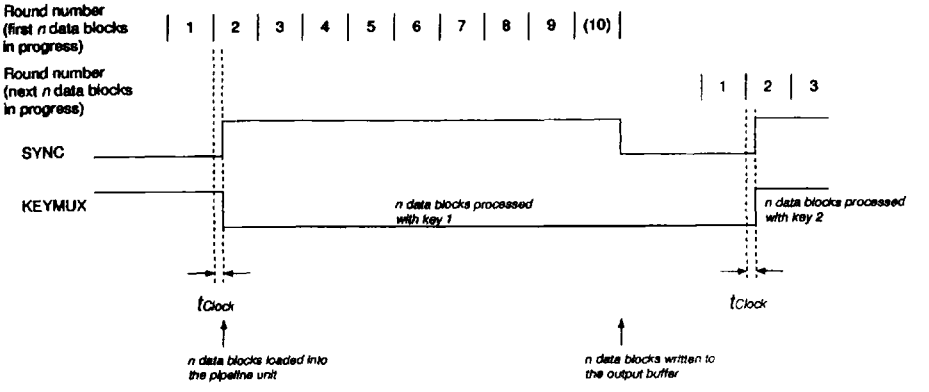
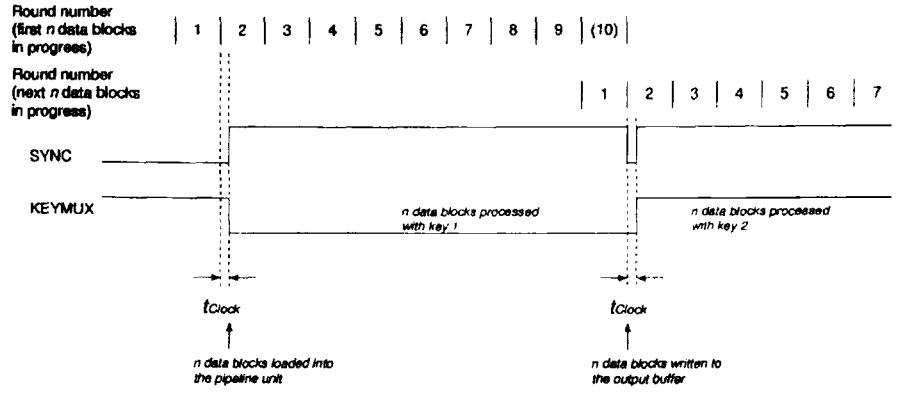


Figure 11. Timing of interface signals at maximum speed



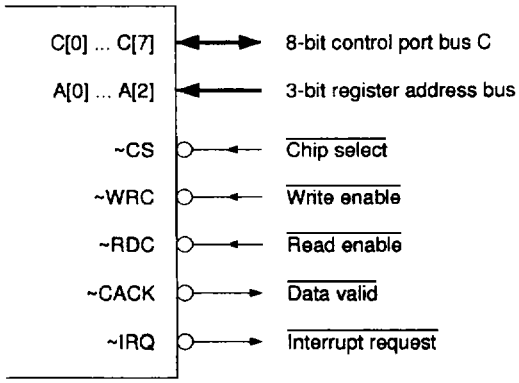
Signal Name	Type	Active	Description
KMUX	O	H	This signal changes its status with the rising edge of the block synchronization signal SYNC. It is low if the *n* data blocks inside the pipeline are converted by means of key 1, and high in case of conversion by means of key 2 (see key select register). The signal is synchronous to the chip clock. Its gets inactive after a master reset or after the command "data reset" (bit 1 of the command register).
SYNC	O	H	The signal gets active one clock cycle later as *n* data blocks are transferred out of the input buffer into the pipeline[1]. The signal will be reset when the same *n* blocks have been transferred into the output buffer. It is synchronous to the chip clock. The signal gets inactive after a master reset or after the command "data reset" (bit 1 of the command register).
CLK2	O	↑	Output signal with half of the chip clock frequency.

1. By this means a minimal pulse length of one clock cycle can be guaranteed, even if the pipeline is loaded and unloaded at the same time.

Control port

Typical microcontrollers for the controlling and the monitoring of the V_{INCI} Chip will be Intel's 8031/8051 and the like. Accordingly, the control port consists of an 8-bit bidirectional data bus and the usual control signals as depicted in Fig. 12.

Figure 12. Control port signals of the V_{INCI} Chip.



Control port signals

Signal Name	Type	Active	Description
C[0] ... C[7]	I/O	H	Bidirectional data bus of the control port. LSB is C[0].
A[0] ... A[2]	I	H	Address lines for control port registers. LSB is A[0].
~WRC	I	L	Write enable signal of port C.
~RDC	I	L	Read enable signal of port C.
~CACK	O	L	Port C's acknowledge signal. In case of a read access it is set if data has been loaded successfully. It is reset after the rising edge of either ~CS or ~RDC. In case of a write access it is set when output data is valid on port C. It is reset after the rising edge of either ~CS or ~WRC.
~CS	O	L	Chip select signal.
~IRQ	O	L	Interrupt request. This signal indicates that an interrupt condition of the interrupt source registers has been satisfied, provided that the corresponding bit of the interrupt enable register is set. The interrupt can be inactived by reading the interrupt source register, by a master reset, or by a reset of the control port register.

All functions of the V_{INCI} Chip can be controlled and monitored via the control port. Seven directly addressable registers are provided and listed on Table 1. The command register and the key command register initialize the desired operations directly. The other registers determine the parameters in both directions or show the status of the chip.

Table 1 : Control port registers

No.	Register	Width	Address A	Access
1	Command register	1 Byte	000	R/W
2	Status register	1 Byte	001	R
3	Key select register	1 Byte	010	R/W
4	Conversion mode register	1 Byte	011	R/W
5	Key command register	1 Byte	100	R/W
6	Interrupt enable register	1 Byte	101	R/W
7	Interrupt status register	1 Byte	110	R

Command register (A=000)

Bit	Function	Initial Value
0	**Start/stop conversion :** This bit permits the control logic to start[1] conversion as soon as at least *n* data blocks are loaded into the input buffer. The conversion itself is defined by the parameters of the conversion mode register and the key select register. During conversion the control logic status bit in the status bit in the status register is set.	0
1	**Data reset :** If this bit is set a running conversion is terminated and data and IV registers are cleared. Neither the settings of the control parameters nor the key registers are influenced.	0
2	**Control port register reset :** The registers 2 ... 6 of the control port are reset to their initial values. The control logic returns to the initial state. Neither data and IV nor the key registers nor status bits 4 and 5 of the status registers are influenced by this command.	0

1. The conversion starts only if the output buffer is empty. Valid data in the output buffer will therefore not be overwritten.

"Data reset" has priority over "start conversion", i.e., conversion will be stopeed and data and IV registers cleared, if both bit 0 and bit 1 are set.

Four different reset commands are supported :

1. A **master reset** clears all data and IV registers asynchronously. Control logic and control port registers are set to their initial values. After a master reset, an off-line selftest is initiated during which all key registers are cleared. A master reset is initiated if the ~ RESET pin is tied to ground.
2. The **data reset** command clears data registers only. A running conversion will be terminated without writing the current results to the output buffer. This reset is synchronous to the chip clock. A data reset is initiated after the command "load encrypted session key". IV registers are not cleared. However, new initialization vectors can be loaded after a data reset.
3. The **control port register reset** does not change neither data nor keys nor the key status. The control logic and the control registers are set to their initial values, except bit 4 and 5 of the status register. This reset is synchronous to the chip clock.
4. The **key reset** command (see key command register) permits termination of the current conversion. A key register reset is initiated during which all key registers and status bits key reset which is synchronous to the chip clock.

Status register (A = 001)

Bit	Function	Initial Value
0	**Control logic status :** 0 : The control logic is waiting in the initial state. 1 : The control logic is active, i.e., a conversion or key operation is running.	0
1	**Input buffer status :** 0 : The input buffer is partially filled or empty. Data can still be loaded. 1 : The input buffer is full.	0
2	**Input buffer status :** 0 : The output buffer is empty. 1 : The output buffer is partially filled or full. Data can still be read.	0
3	**Interrupt status :** 0 : No interrupt pending. 1 : Interrupt pending according to the conditions defined in the interrupt enable register. The bit is reset after reading the interrupt source register.	0
4	**Master key registers R_{MK1} and R_{MK2} status :** 0 : No valid keys in registers R_{MK1} and R_{MK2}. 1 : Valid keys in registers R_{MK1} and R_{MK2}. The master key was successfully loaded by means of the "load master key command".	0
5	**Session key registers R_{SK1} and R_{SK2} status :** 0 : No valid keys in registers R_{SK1} and R_{SK2}. 1 : Valid keys in registers R_{SK1} and R_{SK2}. The session key was successfully loaded by means of the "load encrypted session key command".	0
6	**Off-line selftest status :** 0 : No off-line selftest running. 1 : Off-line selftest running. See also interrupt source register	0
7	**Off-line selftest result :** 0 : Last off-line selftest successfully completed. 1 : Last off-line selftest not successfully completed. See also interrupt source register. This bit if set will only be reset by a master reset or by loading a new master key.	0

Key select register (A=010)

The key select register defines the keys for the conversion of data and **IV** vectors. After the command "start conversion" the first *n* data blocks are converted using key 1, the following alternately with key 2 and key 1 ! The signal KMUX shows which key is selected at the moment. It is left to the user to select the appropriate set of keys. The register is reset after a selftest.

Bit	Function	Initial value
1, 0	**Key 1 :** 0 : Session key register R_{SK1} is selected 1 : Session key register R_{SK2} is selected 2 : Master key register R_{MK1} is selected 3 : Master key register R_{MK2} is selected	0
3, 2	**Key 2 :** 0 : Session key register R_{SK1} is selected 1 : Session key register R_{SK2} is selected 2 : Master key register R_{MK1} is selected 3 : Master key register R_{MK2} is selected	0
5, 4	**Key used for the decryption of IV vectors :** 0 : Session key register R_{SK1} is selected 1 : Session key register R_{SK2} is selected 2 : Master key register R_{MK1} is selected 3 : Master key register R_{MK2} is selected	0

Conversion mode register (A=011)

By means of this register the conversion mode can be defined. The stimulation of the conversion has to be done via the command register using the command "start conversion". After each selftest this register will be set to its initial value.

Bit	Function	Initial value
2, 1, 0	**Conversion mode :** 0 : ECB*n* 1 : CBC*n* 2 : CFB*n* 3 : OFB*n* 4 : MAC	0
3	**Parameter *n* :** 0 : *n* = 1. Data will be loaded, processed, and read out block by block. 1 : *n* = 8. Data will be loaded, processed, and read out in units of eight blocks. . . . . . Do not try to change *n while the input buffer is not empty.*	0
4	**Direction of conversion :** 0 : Data will be encrypted, i.e., internal paths are switched for encryption mode. 1 : Data will be decrypted. Note that no control is given by this command whether a valid set of keys is selected by the key select register.	0
5	**Initialization :** 0 : The first *n* data blocks in the input buffer are regarded as data vectors and will be processed according to the defined conversion mode. 1 : The first *n* data blocks in the input buffer are regarded as initialization vectors and will be processed according to the IV vector status. The bit will be reset after successfully loading of the IV vectors.	0
6	**IV vector status :** 0 : The initialization data are loaded diorectly into the IV register. 1 : The initialization data are converted by ECB mode and afterwards loaded into the IV register. The key is selected according to the status of the key select register.	0

Key command register (A = 100)

Bit	Function	Initial value
0	**Load master key** via the control port : 0 : No operation. 1 : An off-line selftest is initiated. After successful completion,[1] the 16 following bytes supplied to the control port and written to address A = 100 are regarded as master key. The encryption and decryption subkeys are then generated internally and written to the master key registers R_{MK1} and R_{MK2}, respectively. Bit 4 of the status register is set. Parity check is performed during loading if the parity check enable bit is set. After loading of the key the bit will be reset.	0
1	**Load encrypted session key** via the data port : 0 : No operation. 1 : The first two data blocks written to the input buffer are regarded as encrypted session key. The two blocks will be decrypted in ECB mode using the key selected in the key select register. The result is not written to the output buffer. The internal encryption and decryption subkeys will be generated and written to the session key registers R_{SK1} and R_{SK2}, respectively. Bit 5 of the status register will finally be set. Note that a session keyu should only be loaded if the input buffer is empty. After loading of the key the bit will be reset.	0
2	**Master key for the decryption of the session key :** 0 : The master key in register R_{MK1} is selected. 1 : The master key in register R_{MK2} is selected.	0
3	**Parity check enable for master key :** 0 : No parity check on the master key. 1 : During loading of a master key a parity check will be performed. If an error will be detected, an interrupt could be caused.	0
4	**Clear keys registers :** 0 : No operation. 1 : The four key registers R_{MK1}, R_{MK2}, R_{SK1}, R_{SK2}, and bits 4 and 5 of the status register will be cleared.	0

1. The completion can be watched by polling bit 6 of the status register or by interrupt no. 5.

3

Interrupt enable register (A = 101)

This register determines which conditions can cause an interrupt.

Bit	Function	Initial value
0	**Input buffer empty :** This bit masks (0) or enables (1) the input buffer empty interrupt.	0
1	**Output buffer full :** This bit masks (0) or enables (1) the output buffer full interrupt.	0
2	**Parity check error :** This bit masks (0) or enables (1) the parity check error interrupt.	0
3	**Invalid key register selected :** This bit masks (0) or enables (1) the invalid session key register selected interrupt.	0
4	**Error occurred during selftest :** This bit enables (1) or masks (0) the selftest error interrupt.	1
5	**Selftest completed :** This bit masks (0) or enables (1) the selftest completed interrupt.	0

Interrupt status register (A = 110)

This register will be reset in its initial state after it has been read. Bit 0 to 3 will be masked during selftest.

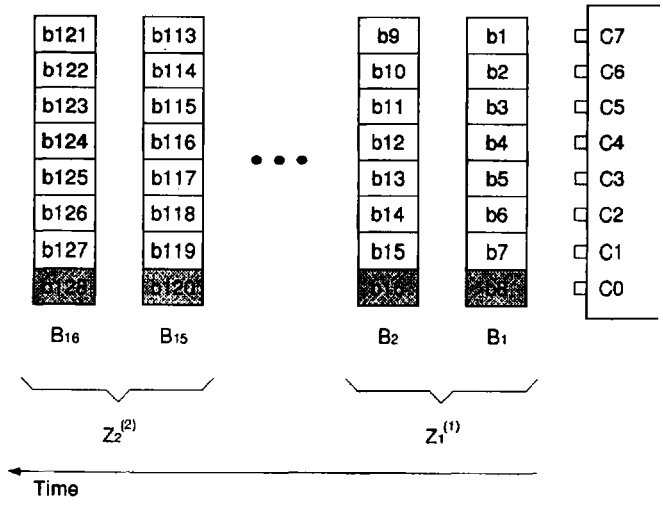
Bit	Function	Initial value
0	**Input buffer empty :** This bit is set if the input buffer has been emptied by the pipeline unit and if bit 0 of the interrupt enable register was set. The bit is being set synchronously to the rising edge of the SYNC signal.	0
1	**Output buffer full :** This bit is set if the output buffer has been filled by the pipeline unit and if bit 1 of the interrupt enable register was set. The bit is being set synchronously to the falling edge of the SYNC signal	0
2	**Parity check error :** This bit is set if a parity check error occurs during the loading of the master key and bit 2 of the interrupt enable register is set.	0
3	**Invalid key register selected :** This bit is set if an invalid session key was selected with the key select register (i.e. bit 5 of the status register was not set) and bit 3 of the interrupt enable register was set.	0
4	**Error occurred during selftest :** This bit is set if an error has occurred during selftest and bit 4 of the interrupt enable register was set.	0
5	**Selftest completed :** This bit is set if the off-line selftest has completed and bit 5 of the interrupt enable register was set.	0

Key Definitions

Loading the master key

The 128-bit master key is transmitted to the control port according to the data representation for I_{DEA}. Z is loaded using sixteen byte transfers. The master key is composed of the first eight 16-bit wide subkeys [2]. The first subkey $Z_1^{(1)}$ is transmitted first. The first transmitted byte is the MSByte B_1, see Fig. 13.

Figure 13. Master key transfer to the V_{INCI} Chip



Optional parity check on the master key

When loading the master key a parity check can be performed on it by setting bit 5 of the key command register. The parity bit completes each 7-bit vector to odd parity. In fig. 13 the parity bits are each marked by a grey pattern.

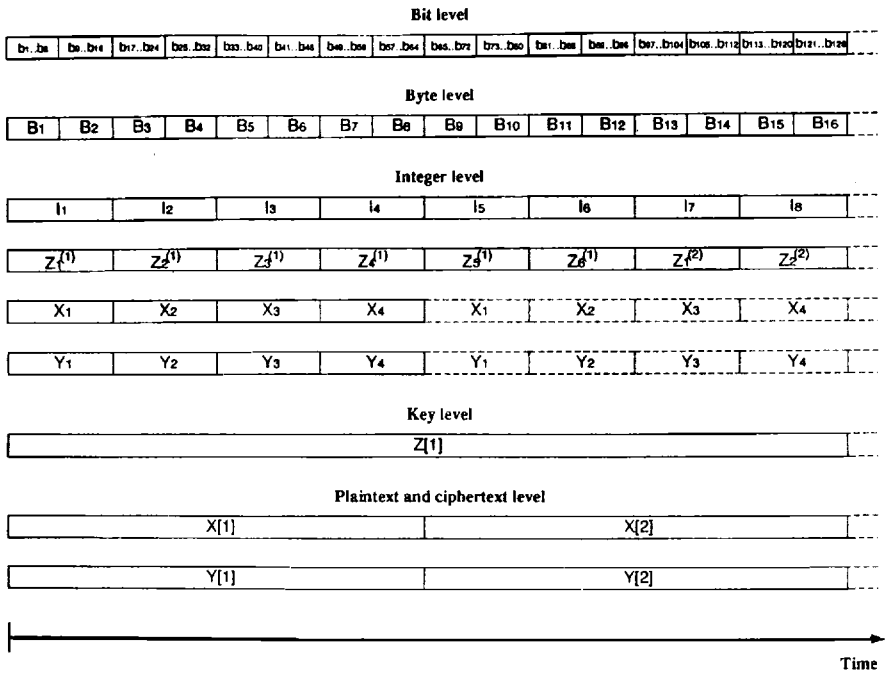
Loading the session key

The loading scheme of the session key via the data port is identical to the loading scheme of the master key via the control port. However, the width of the data port is sixteen bit. Therefore only eight transfers are required. $Z_1^{(1)}$ (b_1 ... b_{16}, b_{16} at pin DB0) is transmitted first.

Data representation for I_{DEA}

In Fig. 14 the data representation for the IDEA is shown on different levels of abstraction. b1, B1, Z[1], and Y[1], respectively, is transmitted.

Figure 14. Data representation for I_{DEA}



The following relations hold :

$$B_i = \sum_{k=0}^{7} b_{8i-k}2^k$$

$$Ii = B_{2i-1}2^8 + B_{2i}$$

$$Z[i] = \sum_{k=0}^{7} I_{8i-k}2^{16k}$$

$$X[i] = \sum_{k=0}^{3} I_{4i-k}2^{16k}$$

$$Y[i] = \sum_{k=0}^{3} I_{4i-k}2^{16k}$$

Ratings and Characteristics

Absolute Maximum Ratings

Ambient temperature ..... 0 °C to + 70 °C
Storage temperature ..... −65 °C to + 150 °C
Voltage on any pin with respect
to ground ..... V_{SS} −0.5 V to V_{DD} + 0.5 V
Power dissipation ..... 1.0 W

DC and Operating Characteristics

T_A = + 25 °C, V_{DD} = + 5 V ± 10 %, V_{SS} = 0 V

Symbol	Parameter	Limits		Unit	Test Conditions
		Min	Max		
V_{DD}	Power supply				
V_{IL}	Low Level Input Voltage				
V_{IH}	High Level Input Voltage				
V_{OL}	Low Level Output Voltage				
V_{OH}	High Level Output Voltage				
I_{IL}	Input Leakage Current				
I_{DD}	Total Supply Current				
C_{OUT}	Output Capacitance				
C_{IN}	Input Capacitance				
$C_{I/O}$	I/O Capacitance				

3

AC Characteristics

T_A = + 25 °C, V_{DD} = + 5 V ± 5 %, V_{SS} = 0 V, F_{Clk} = 33 MHz

Read and Write Operations

Symbol	Parameter	Limits		Unit
		Min	Max	
T_{AS}	Setup Time	...		ns
T_{AH}	Hold Time	...		ns
T_{RS}	Setup Time	...		ns
T_{RH}	Hold Time	...		ns
T_{PS}	Setup Time	...		ns
T_{PH}	Hold Time	...		ns
T_{SS}	Setup Time	...		ns
T_{SH}	Hold Time	...		ns
T_{AD}	Delay Time	...		ns
T_{AO}	Valid to High	...		ns
T_{AU}	High to Invalid	...		ns
T_{AZ}	High to High Impedance		...	ns
T_{DS}	Setup Time	...		ns
T_{DH}	Hold Time	...		ns
T_{DD}	Delay Time	...		ns
T_{DU}	Valid to Invalid	...		ns
T_{DZ}	Valid to High Impedance		...	ns

Other Timings

Symbol	Parameter	Condition	Min	Typ	Max	Unit
F_{CLOCK}	Clock Frequency		0.2	25	33	MHz
T_{CP}	Clock Period		30	40	5×10^3	ns
T_{CH}	Clock High Time	F_{CLOCK} = 25 MHz	20	21	23	ns
T_{CL}	Clock Low Time	F_{CLOCK} = 25 MHz	17	19	20	ns
T_{CS}	Clock Skew	CLK2 to CLOCK	– 11	0	7	ns

Mechanical Data

Pin Assignments (by functions)

Symbol	Pin	Type	Description
CLOCK	N12	I	Master clock : Chip clock with positive edge active.
Clk2	J3	O	Output signal with half of the CLOCK'(s frequency. Can be used to drive CLKA and CLKB, respectively.
~RESET	L12	I	Master reset. If this signal is tied low, all data registers will be cleared and the control port registers will be initialized. The control logic will be set to its initial state. After master reset, an off-line self-test is initiated. Active low schmitt trigger input.
KMUX	N1	O	This signal changes its state with the rising edge of the bloci synchronization signal SYNC. It is low it the *n* data blocks inside the pipline are converted by means of key 1, and high in case of conversion by means of key 2 (see key select register). The signal will be reset after a master reset of after the command "data reset" (bit 1 of the command register).
SYNC	M2	O	This signal goes high one clock cycle later as *n* data blocks are transferred into the pipeline. The signal will be reset when the same n blocks have been transferred into the output buffer or after a master reset or the command "data reset" (bit 1 of the command register).
~CS	N9	I	Chip select signal, active low.
~RDC	M11	I	Read enable signal for the control port C, active low.
~WRC	P14	I	Write enable signal for the control port C, active low.
~CACK	G2	O	Port C's acknowledge signal. In case of a read access it is set after data has been loaded successfully. It is reset after the rising edge of ~RDC. In case of a write access it is set when output data is valid on port C. Is is reset after the rising edge of ~WRC.
C[0] C[1] C[2] C[3] C[4] C[5] C[6] C[7]	L13 K12 K14 J13 H14 H13 G13 G14	I/O I/O I/O I/O I/O I/O I/O I/O	Bidirectional data bus of the control port. LSB is C[0].
A[0] A[1] A[2]	P13 M10 N11	I I I	Address lines for selecting control port registers. LSB is A[0].
~IRQ	L1	O	Interrupt request. The signal indicates that an interrupt condition of the interrupt source registers has been
CLKA	P8	I	Input data port A's clock signal with positive edge active.
~WRA	M13	I	Input data port A's write enable signal. If this signal is low, the data on DA[0] ... DA[15] is fetched by port A on the next rising edge of CLKA.
~WAITA	J1	O	Port A's wait signal is set if data has been loaded but not transferred yet to the input buffer.
DIR	L2	O	"Data-in-ready" is active until the input buffer is completely filled. It is set after a master reset and after the command "data reset" (bit 1 of the command register). This signal is inactive during selftest.

Pin Assignments (by functions) (continued)

Symbol	Pin	Type	Description
DA[0] DA[1] DA[2] DA[3] DA[4] DA[5] DA[6] DA[7] DA[8] DA[9] DA[10] DA[11] DA[12] DA[13] DA[14] DA[15]	F12 D14 C14 D13 E12 B14 C13 D12 B12 C11 A12 B11 C10 A10 B9 B8	I I I I I I I I I I I I I I I I	Input data port A : Data bus for input data, initialization vectors, and session keys. LSB is DA[0].
CLKB	P10	I	Output data port B's clock signal with positive edge active.
~RDB	N14	I	Output data port B's read enable signal. If this signal is driven low, output data is available on DB[0] ... DB[15] after the next rising edge of CLKB.
~WAITB	H2	O	Port B's wait signal is set when a read request is detected on port B but the next data word has not been read out yet from the output buffer.
DOR	M1	O	"Data-out-ready" is active until the output buffer is empty. It is reset after a master reset and after the command "data reset" (bit 1 of the command register). This signal is inactive during selftest.
DB[0] DB[1] DB[2] DB[3] DB[4] DB[5] DB[6] DB[7] DB[8] DB[9] DB[10] DB[11] DB[12] DB[13] DB[14] DB[15]	A6 A5 B5 C5 A2 B3 C4 A1 B1 C2 D3 C1 D2 E3 E1 F2	O O O O O O O O	Output data port B : Tristate data bus for output data. LSB is DB[0].
StRet	L3	O	Start of retention
IERet	M8	I	Internal/external retention signal, internally pulled-up
TCLK	N4	I	Boundary scan clock
TRST	P4	I	Boundary scan reset, active-low, internally pulled-up
TDI	M6	I	Boundary scan serial input, internally pulled-up
TDO	P3	O	Boundary scan serial output
TMS	N5	I	Boundary scan control, internally pulled-up
STIni	P6	O	Initiate selftest with active rising edge, internally pulled-down
STStat	P2	O	Status of selftest
STRes1 STRes2	N3 M4	O O	Concurrent selftest result monitor, bit 1 Concurrent selftest result monitor, bit 2

Pin Assignments (by functions) (continued)

Symbol	Pin	Type	Description
VDD	A11 A13 B2 B4 C3 G1 M7 M12 N10 N13	P P P P P P P P P P	Power supply : + 5 V power input
VSS	A14 B7 C8 C12 F14 H1 K3 M3 M14 P1	P P P P P P P P P P	Ground : 0 V input

3

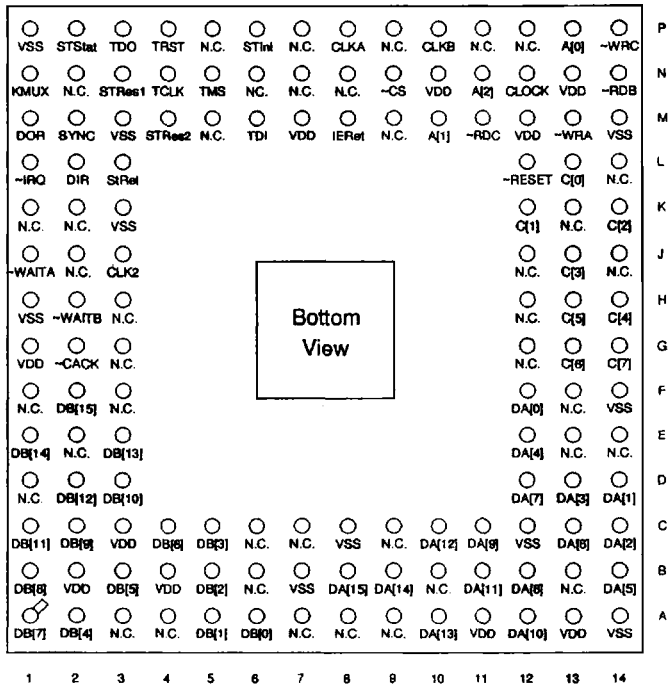
Pin assignment (by order)

Pin	Signal	Typ	Pin	Signal	Typ
A1	DB[7]	O	H1	VSS	P
A2	DB[4]	O	H2	~WAITB	O
A3	N.C.	-	H3	N.C.	-
A4	N.C.	-	H12	N.C.	-
A5	DB[1]	O	H13	C[5]	I/O
A6	DB[0]	O	H14	C[4]	I/O
A7	N.C.	-	J1	~WAITA	O
A8	N.C.	-	J2	N.C.	-
A9	N.C.	-	J3	CLK2	O
A10	DA[13]	I	J12	N.C.	-
A11	VDD	P	J13	C[3]	I/O
A12	DA[10]	I	J14	N.C.	-
A13	VDD	P	K1	N.C.	-
A14	VSS	P	K2	N.C.	-
B1	DB[8]	O	K3	VSS	P
B2	VDD	P	K12	C[1]	I/O
B3	DB[5]	O	K13	N.C.	-
B4	VDD	P	K14	C[2]	I/O
B5	DB[2]	O	L1	~IRQ	O
B6	N.C.	-	L2	DIR	O
B7	VSS	P	L3	StRet	O
B8	DA[15]	I	L12	~RESET	I
B9	DA[14]	I	L13	C[0]	I/O
B10	N.C.	-	L14	N.C.	-

Pin assignment (by order) (continued)

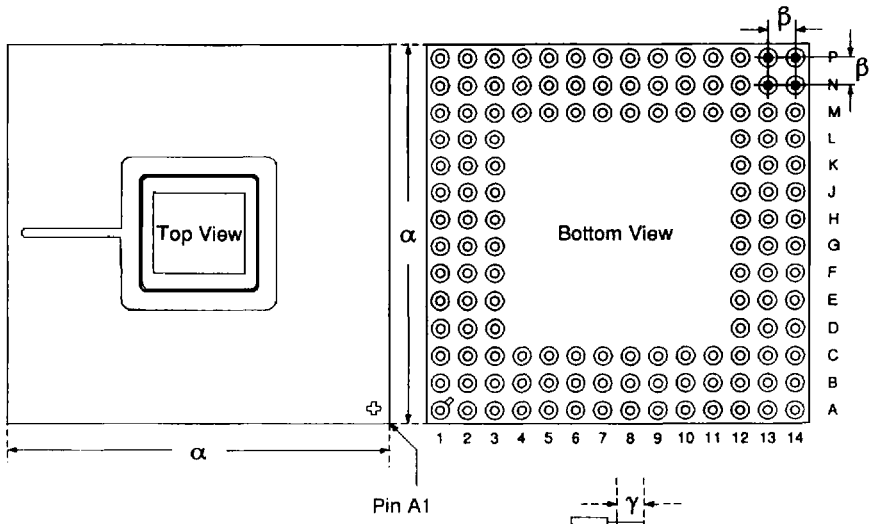
Pin	Signal	Typ	Pin	Signal	Typ
B11	DA[11]	I	M1	DOR	O
B12	DA[8]	I	M2	SYNC	O
B13	N.C.	-	M3	VSS	P
B14	DA[5]	I	M4	STRes2	O
C1	DB[11]	O	M5	N.C.	-
C2	DB[9]	O	M6	TDI	I
C3	VDD	P	M7	VDD	P
C4	DB[6]	O	M8	IERet	I
C5	DB[3]	O	M9	N.C.	-
C6	N.C.	-	M10	A[1]	I
C7	N.C.	-	M11	~RDC	I
C8	VSS	P	M12	VDD	P
C9	N.C.	-	M13	~WRA	I
C10	DA[12]	I	M14	VSS	P
C11	DA[9]	I	N1	KMUX	O
C12	VSS	P	N2	N.C.	-
C13	DA[6]	I	N3	STRes1	O
C14	DA[2]	I	N4	TCLK	I
D1	N.C.	-	N5	TMS	I
D2	DB[12]	O	N6	N.C.	-
D3	DB[10]	O	N7	N.C.	-
D12	DA[7]	I	N8	N.C.	-
D13	DA[3]	I	N9	~CS	I
D14	DA[1]	I	N10	VDD	P
E1	DB[14]	O	N11	A[2]	I
E2	N.C.	-	N12	CLOCK	I
E3	DB[13]	O	N13	VDD	P
E12	DA[4]	I	N14	~RDB	I
E13	N.C.	-	P1	VSS	P
E14	N.C.	-	P2	STStat	O
F1	N.C.	-	P3	TDO	O
F2	DB[15]	O	P4	TRST	I
F3	N.C.	-	P5	N.C.	-
F12	DA[0]	I	P6	STIni	O
F13	N.C.	-	P7	N.C.	-
F14	VSS	P	P8	CLKA	I
G1	VDD	P	P9	N.C.	-
G2	~CACK	O	P10	CLKB	I
G3	N.C.	-	P11	N.C.	-
G12	N.C.	-	P12	N.C.	-
G13	C[6]	I/O	P13	A[0]	I
G14	C[7]	I/O	P14	~WRC	I

Figure 15. 132-Pin Grid Array



Package Dimensions

Figure 16. Dimensions of PGA 132



Dim	mm	inches
α	35.6	1.400
β	2.54	0.100
γ	3.30	0.130
δ	0.457	0.018
ε	7.34	0.289