ORCHIDĒE
SEMICONDUCTOR

TRIPLE-DES/Data Management Synthesizable IP Core DES-DMGR

1. Block Description

The Orchidée DES-DMGR Intellectual Property (IP) Core is a Verilog HDL based intelligent microprocessor peripheral designed to implement the NBS FIPS Pub. 46-2 (Data Encryption Standard) with Triple-DES, Cipher Block Chaining (CBC) encryption and decryption capability, while managing block data I/O via microprocessor read/write and interrupt signals. The core has been designed to off-load the microprocessor of the numerous interrupts encountered while interfacing with typical DES hardware. Thus, the DES-DMGR is ideal for applications requiring high data throughput in intensive interrupt or multitasking environments or those with long interrupt latencies. The DES-DMGR includes the following features:

- Electronic Code Book (ECB), CBC, Triple-DES and Triple-DES/CBC modes of operation.
- Independent, memory-mapped, input and output data queues (fifo's).
- Independently parameterized fifo depths.
- Fifo data management control signal maskable interrupts.
- Parameterized 32 or 64-bit synchronous bus interface.
- Memory-mapped control/status registers.
- DES key parity checking.
- Single, super synchronous clock domain.
- 20 clock cycle encryption/decryption.

The DES-DMGR is command-oriented. It supports both burst mode and controlled queue and de-queue of information from I/O fifo blocks for data transfers. Upon achieving a programmable fullness/emptiness level in the output/input fifo, the microprocessor is notified via an interrupt signal which is automatically reset upon a read of the interrupt status register.

The fifo blocks in the DES-DMGR contain dual-port RAMs which may be implemented either with discrete cells (via synthesis) or with compiled dual-port RAMs with the desired density. These dual-port RAMs must have one asynchronous write port (with write enable low signal) and one asynchronous multiplexed read port for proper operation in the circuit.

2. Block Diagram

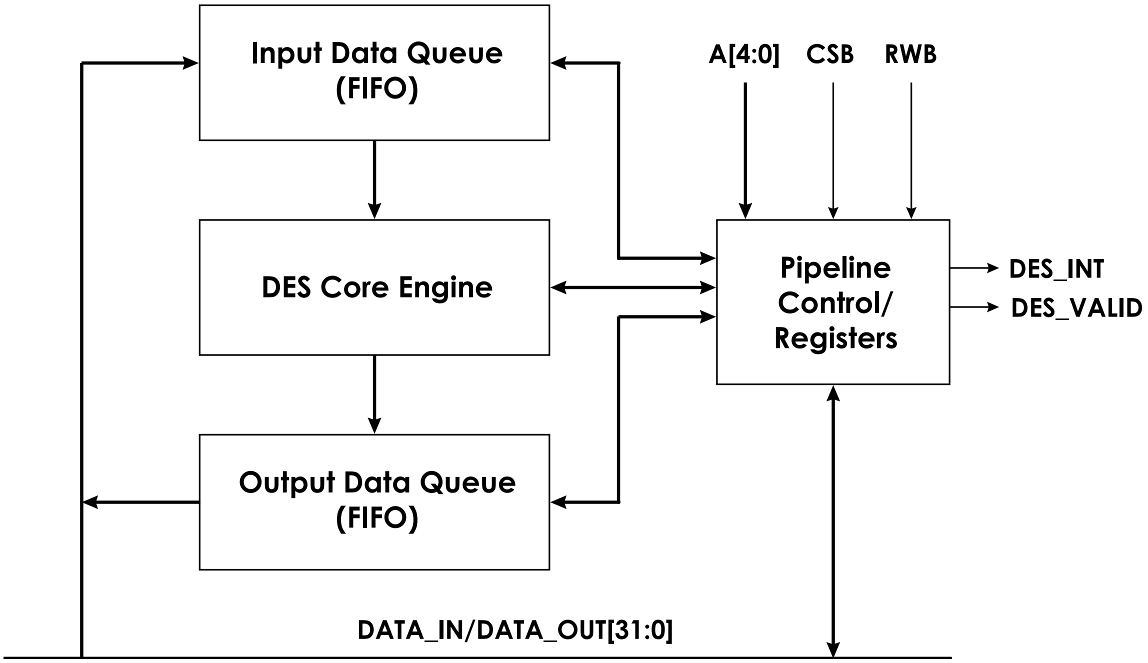

Figure 1: DES-DMGR Block Diagram.

The DES-DMGR acts as a data pipeline from input data queue through the DES core to output data queue. Clear text or cipher data is first queued in the input fifo before processing in the DES core. Resultant cipher or clear text is then stored in the output fifo.

The pipeline is controlled via command and status registers which appear in the microprocessor memory map, thus permitting ease of programming and control of the peripheral.

3. Pin List

Name	Type	Description
A[4:0]	input bus	5-bit register address bus
CSB	input	Active low chip select
RSTB	input	Global active low asynchronous reset
CLK	input	Clock input
RWB	input	Read/write-low input signal
DATA_IN[31:0]	input bus	Microprocessor data bus input
DATA_OUT[31:0]	output bus	Microprocessor data bus output
DES_VALID	output	Indicates valid data present on data output bus during DES-DMGR accesses. Active high.
DES_INT	output	Indicates a valid interrupt present in the interrupt status register. Active high.

Table 1: DES-DMGR Pin I/O List.

4. Microprocessor Interface Bus Timing

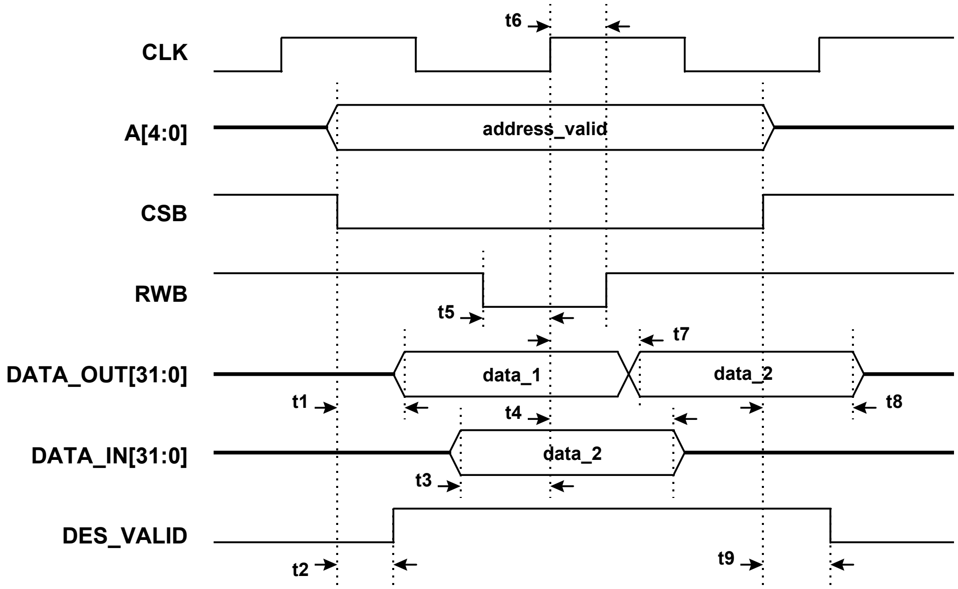

Figure 2: Microprocessor Interface Bus Timing Diagram.

4. Microprocessor Interface Bus Timing (cont.)

Name	Description	Min	Typ	Max
t1	Address valid, CSB low to DATA_OUT valid			TBD
t2	Address valid, CSB low to DES_VALID high			TBD
t3	DATA_IN setup time	TBD		
t4	DATA_IN hold time	TBD		
t5	RWB low to rising edge of CLK	TBD		
t6	Rising edge of clock to RWB high	TBD		
t7	Rising edge of clock to DATA_OUT valid			TBD
t8	Address invalid, CSB high to DATA_OUT invalid			TBD
t9	Address invalid, CSB high to DES_VALID low			TBD

Table 2: DES-DMGR Microprocessor Interface Timing.

5. Register Address Map

Name	Address	Type	Reset Value	Description
isr	0x00	RO	0x00000000	Interrupt status register
imr	0x01	RW	0x0000007f	Interrupt mask register
cmd	0x02	WO	N/A	Command register
mod	0x03	RW	0x00000000	Mode register
fcounter_iq	0x04	RW	0x00000000	Input counter register
fcounter_oq	0x05	RW	0x00000000	Output counter register
key1l	0x06	RW	0x00000000	DES key1 lsb register
key1m	0x07	RW	0x00000000	DES key1 msb register
key2l	0x08	RW	0x00000000	DES key2 lsb register
key2m	0x09	RW	0x00000000	DES key2 msb register
key3l	0x0a	RW	0x00000000	DES key3 lsb register
key3m	0x0b	RW	0x00000000	DES key3 msb register
ivecl	0x0c	RW	0x00000000	DES initial vector lsb register
ivecm	0x0d	RW	0x00000000	DES initial vector msb register
data_inl	0x0e	WO	N/A	Data_in lsb
data_inm	0x0f	WO	N/A	Data_in msb
data_outl	0x10	RO	0xXXXXXXXX	Data_out lsb
data_outm	0x11	RO	0xXXXXXXXX	Data_out msb

Note: RW = Read/Write; RO = Read Only; WO = Write Only;

Table 3: Register Address Map.

6. Register Bit-Level Description

6.1. Interrupt Status Register: isr

Bit(s)	[31:8]	[7]	[6]	[5]	[4]	[3]	[2]	[1]	[0]
Name			PER	XDO	XFH	RFH	XDU	RFO	FIN
Reset Value	0	0	0	0	0	0	0	0	0

Description:
FIN: The operation has terminated normally (encryption/decryption complete) after an XME command.
RFO: The pipeline has stalled due to an output data queue (fifo) full condition.
XDU: The pipeline has stalled due to an input data queue (fifo) empty condition.
RFH: The output data queue (fifo) has filled to the value indicated in the fcounter_oq register. Pipeline has not stalled.
XFH: The input data queue (fifo) has emptied to the level indicated in the fcounter_iq register. Pipeline has not stalled.
XDO: A write was attempted to the input data queue (fifo) when the fifo was already full.
PER: A DES key parity error condition exists in one of the 3 DES keys contained in the key registers.

Note: All unmasked interrupts are automatically cleared upon a read of the isr.

6.2. Interrupt Mask Register: imr

Bit(s)	[31:8]	[7]	[6]	[5]	[4]	[3]	[2]	[1]	[0]
Name			PER	XDO	XFH	RFH	XDU	RFO	FIN
Reset Value	0	0	1	1	1	1	1	1	1

Description:
Writing a 1 to any of the register bits causes it's associated interrupt in the isr to be masked. The interrupt is generated as soon as it becomes unmasked.

6.2. Interrupt Mask Register: imr (cont.)

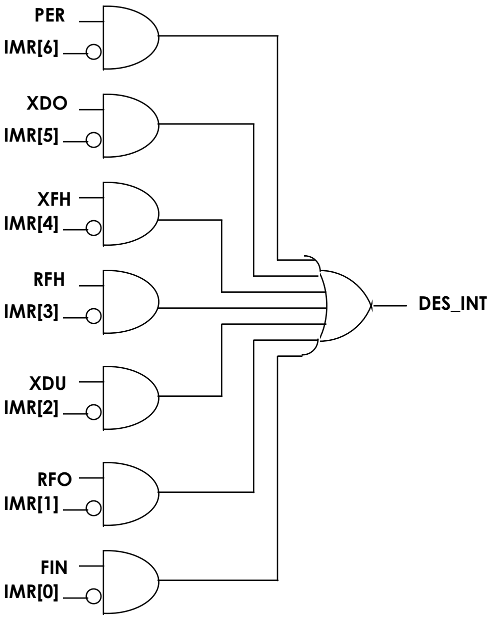

Figure 3: DES_INT Logical Equivalent.

6. Register Bit-Level Description (cont.)

6.3. Command Register: cmd

Bit(s)	[31:8]	[7]	[6]	[5]	[4]	[3]	[2]	[1]	[0]
Name							XHF	XME	RST
Reset Value	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A

Description:
RST: causes a software reset of the input data queue, output data queue, DES core and all associated state machines. Any pending interrupts will be cleared as a result of this command.
XME: causes the DES-DMGR to begin operation (encryption/decryption) on data contained in the input data queue (fifo) and to generate a FIN interrupt upon successful completion of encryption or decryption of all of the data contained in it. This command is used if all of the data to be encrypted/decrypted can be written to the input fifo in a single burst (i.e. burst mode). Note that RFO and RFH interrupts are still valid should their associated conditions become valid in the output data queue (fifo) and they are not masked.
XHF: causes the DES-DMGR to begin operation (encryption/decryption) on data contained in the input data queue (fifo) and to generate an underun interrupt (XDO) if this fifo becomes empty. This command is used when encryption or decryption is required on blocks of data longer than the input fifo is deep. When this command is issued, the XFH interrupt is normally used to manage the re-filling of data into the input queue when it empties to the value designated in the fcounter_iq register and to avoid a stall in the data pipeline. Upon the final write of data to the input data queue, an XME command is issued such that a FIN interrupt may be generated and the operation may terminate normally.

Note: Only 1 command should be issued at any time. Should more than 1 command be issued, the DES-DMGR will respond to the command with the lowest bit value (i.e. RST takes priority over XME and XHF, XME takes priority over XHF).

6. Register Bit-Level Description (cont.)

6.4. Mode Register: mod

Bit(s)	[31:8]	[7]	[6]	[5]	[4]	[3]	[2]	[1]	[0]
Name							CBC	MOD	OP
Reset Value	0	0	0	0	0	0	0	0	0

Description:
OP: A 0 value indicates a decryption operation. A 1 value indicates an encrypt operation.
MOD: A 0 value indicates normal (i.e. not triple) DES mode. A 1 value indicates triple-DES mode.
CBC: A 0 value indicates ECB (i.e. not CBC) mode. A 1 value indicates CBC mode. Note that this bit must be toggled to begin CBC data processing on new blocks of data which require use or reuse of the initial vector (ivec) after performing CBC operations on a previous block of data. At least 2 clock cycles in between toggles of this bit is required for proper reset of CBC mode.

6.5. Input Data Queue Counter Register: fcounter_iq

	Bits[31:0]
Name	fcounter_iq[31:0]
Reset Value	0x00000000

Description:
The fcounter_iq register is used to determine the XFH interrupt. When the input data queue (fifo) has been filled to a value higher than the value indicated in the fcounter_iq register, the XFH interrupt will then be generated when the fifo empties to the value indicated in this register. Note that the XFH interrupt will not be regenerated until the fifo is again filled to a level higher that the value stored in the fcounter_iq register and subsequently re-emptied to this value.

6. Register Bit-Level Description (cont.)

6.6. Output Data Queue Counter Register: fcounter_oq

	Bits[31:0]
Name	fcounter_oq[31:0]
Reset Value	0x00000000

Description:
The fcounter_oq register is used to determine the RFH interrupt. The RFH interrupt is generated when the output data queue (fifo) fills to the value indicated in this register. Note that the RFH interrupt will not be regenerated until the output fifo has been first emptied to a level below the value stored in the fcounter_oq register and subsequently refilled to this value.

6.7. KEY and IVEC Registers

	Bits[31:0]
Name	key1l[31:0]
Name	key1m[31:0]
Name	key2l[31:0]
Name	key2m[31:0]
Name	key3l[31:0]
Name	key3m[31:0]
Name	ivecl[31:0]
Name	ivecm[31:0]
Reset Value	0x00000000

Note: key1 is used for normal (i.e. not triple) DES operations.

6. Register Bit-Level Description (cont.)

6.7. DATA_INL and DATA_INM

	Bits[31:0]
Name	data_inl[31:0]
Name	data_inm[31:0]
Reset Value	N/A

Description:
The data_inl bits are the least significant bits of the 64-bit input word to be operated on (encrypted or decrypted) by the DES core. The data_inm bits are the most significant bits of the 64-bit input word to be operated on (encrypted or decrypted) by the DES core. Since the input data queue fifo counter increments on writes to the most significant bits, always write the data_inl word first before the data_inm word for proper operation.

6.7. DATA_OUTL and DATA_OUTM

	Bits[31:0]
Name	data_outl[31:0]
Name	data_outm[31:0]
Reset Value	0xXXXXXXXX

Description:
The data_outl bits are the least significant bits of the 64-bit resultant output word from the DES core. The data_outm bits are the most significant bits of the 64-bit output word from the DES core. Since the output data queue fifo counter decrements on reads from the most significant bits, always read the data_outl word first before the data_outm word for proper operation.

7.0. DES CORE THEORY OF OPERATION

7.1. ECB Mode (Non Triple-DES)

The DES core module of the DES_DMGR core operates synchronously using the positive edge of the clock (CLK) input. An asynchronous reset (RSTB, active low) is provided to the entire module which causes a reset.

After a reset, the DES module is ready to begin either an encryption or decryption operation (depending on the value of the OP bit in the mod register) upon the first rising edge of the CLK after an XHF or XME command. During a normal (i.e. non triple-DES) operation the key1 register value is used as the DES key in the cipher algorithm. It is imperative that the key registers not be changed for the entire duration of any encrypt/decrypt operation. They may be changed only after an FIN interrupt has been received by the microprocessor or after either a hardware or software reset (RST command).

The resultant data stored in the output data queue will be:

Encrypt: Cipher(n) = E[key1, Data(n)];
Decrypt: Data(n) = D[key1, Cipher(n)];

n = 1, 2, 3…;
E = Encrypt;
D = Decrypt;

7.2. CBC Mode (Non Triple-DES)

During CBC mode, the contents of the ivec register is used only during the first encrypt/decrypt operation after an XHF or XME command, indicating the first block of data in a CBC mode operation. Note the ivec register is used only on the first 64-bit block of data input. Subsequent operations while there is valid data in the input data queue will use the resultant cipher or clear text from the previous operation for XORing with input or output data for encryption or decryption operations, respectively. To begin an operation on a new block of data with a new ivec value, the CBC bit of the mod register must be brought low for at least 2 CLK cycles and then re-asserted high for this new ivec value to be used. Note the contents of the ivec register, like the key registers, must also not be changed for the entire encrypt/decrypt operation.

The resultant data stored in the output data queue during CBC mode will be:

Encrypt: Cipher(n) = E[key1, (Data(n) + Cipher(n-1))];
Decrypt: Data(n) = D[key1, Cipher(n)] + Data(n-1);

n = 1, 2, 3…; for n = 1, Cipher(0) = Data(0) = ivec;
E = Encrypt;
D = Decrypt;
+ = Exclusive OR

7.3. ECB Mode (Triple-DES)

Setting the MOD bit to a 1 in the mod register when the encryption/decryption operation begins will cause a triple-DES operation to occur which will make use of the additional key2 and key3 registers. The equations for triple encrypt and decrypt operations is as follows:

Encrypt: Cipher(n) = E[key3, D[key2, E[key1, Data(n)]]]
Decrypt: Data(n) = D[key1, E[key2, D[key3, Cipher(n)]]]

n = 1, 2, 3…;
E = Encrypt;
D = Decrypt;

Operations furthest inside of the brackets are implemented first.

7.4. CBC Mode (Triple-DES)

Triple-DES in CBC mode takes the following format:

Encrypt: Cipher(n) = E[key3, D[key2, E[key1, (Data(n) + Cipher(n-1))]]];
Decrypt: Data(n) = D[key1, E[key2, D[key3, Cipher(n)]]] + Data(n-1);

n = 1, 2, 3…; for n = 1, Cipher(0) = Data(0) = ivec;
E = Encrypt;
D = Decrypt;
+ = Exclusive OR

Operations furthest inside of the brackets are implemented first.

8.0. DES Data Manager Pipeline State Machine

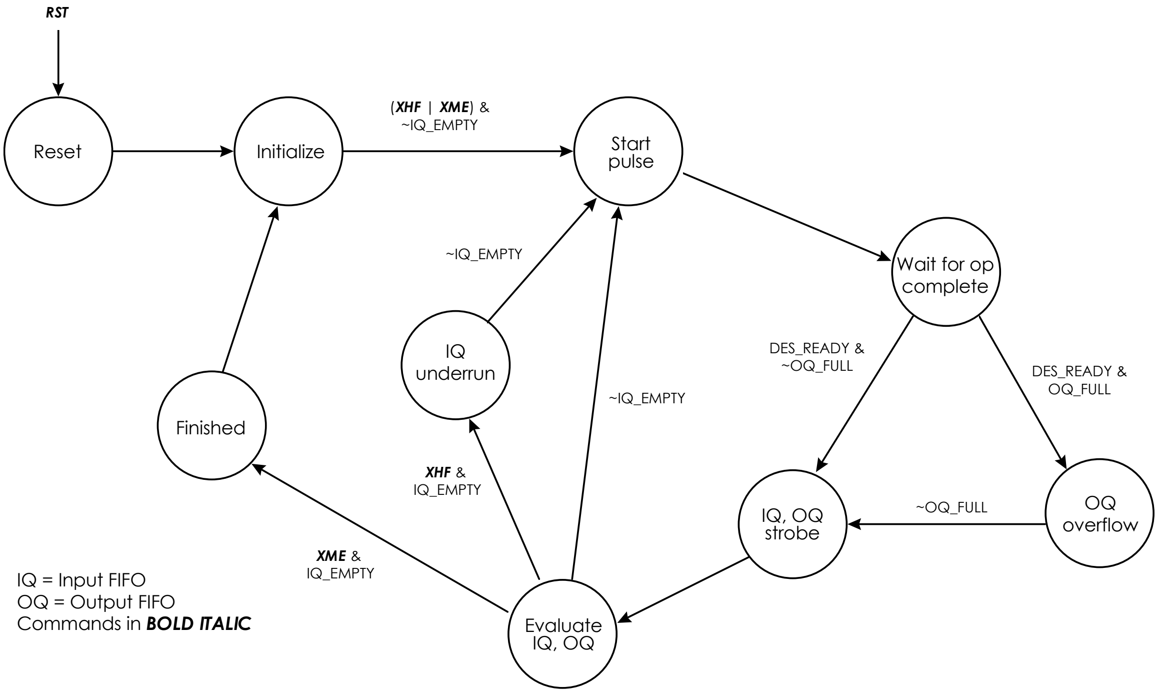

Figure 4: DES Data Manager Pipeline State Machine.

The DES Data Manager data pipeline is controlled by a universal state machine which governs reads and writes from the input data queue (IQ FIFO) and output data queue (OQ FIFO) as well as the necessary strobe signals used in the DES core. Here, commands (as input into the command register) take priority over the state of the input and output FIFO's. Global software reset takes top priority at any time by issuing an RST command.

Contact Information

Company Headquarters:

Orchidée Semiconductor Sàrl
Chemin de la Rosière 40
CH-1012 Lausanne, Switzerland
Telephone: +41 21 711 0687
Fax: +41 1 355 3187
http://www.orchidée.com

USA Sales:

Northern California Technical Sales, Inc.
1762 Technology Drive, Suite 204
San Jose, CA 95110 USA
Telephone: 408-327-0540
http://www.e-rep.com

European Sales:

Hake-Mechatronik
Ruhbronnweg 11/1
74385 Pleidelsheim, Germany
Telephone +49 (7144) 88 4550
Fax: +49 (7144) 88 4551
http://www.rainer-hake.de