CHAPTER 4
SYSTEM DEVELOPMENT

This chapter describes the concerned theories that are used for the "**Die attach machine error detection**" system development. It starts from the basic concepts of die attach process and machine. Following sections are details of the international standard association and the standard that relates to semiconductor equipment communications. The communications capabilities of the die attach machine are also described at the end of this chapter with some basic concepts on the Error detection system.

4.1 Die attach machine and procedure

There are many suppliers and models of die attach machine. However, not every machine can interface and communicate with the host. Some machines are designed for the purpose of die attach only whereas others extend their capabilities on the host communication interface. However, the host communication interface requires advanced technology for system development. Hence, some suppliers are not capable enough for this technology now.

ESEC SA Inc. is a leader of die attach machine suppliers. Many models of die attach machine are produced and developed by this company. The latest model is 2007 that is an advanced technology, high precision machine for picking up the circuit chips (die) from a sawn wafer and bonds them onto leadframes.

It is not only the advanced technology in this model but also adding the host communication interface into the machine. The machine is controlled by a built-in computer system so that the host communication interface is possible. It expands the machine capabilities to be higher than its competitors. Hence, the ESEC die attach machine model 2007 is selected for the "**ERROR DETECTION SYSTEM**" development.

The figure 4.1 shows the ESEC die attach machine model 2007 and the front panel of the machine.

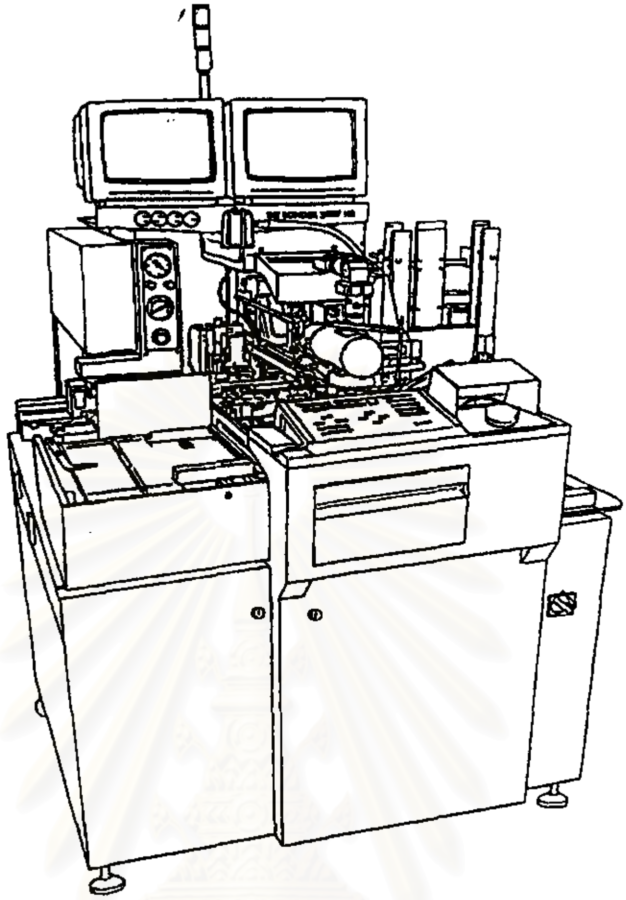

Figure 4.1: ESEC2007 operator controls. (Front side)

4.1.1 Die attach operations

Due to the research related to detection of machine states and errors on the machine, some parts and basic machine operation have to be known. The die attach operations relate to picking the sawn chip (die) up and then place onto the leadframes. There are some important parts that directly relate to this machine operation. The first part is bond module, which consist of a bond arm and bond head. This part is used to pick the sawn chip (die) from the wafer and then move to bond on the leadframes. Another part is leadframes indexing module that is a part to transfer leadframes to epoxy dispensing and bond location. The following figures illustrate the bond module and indexing module.

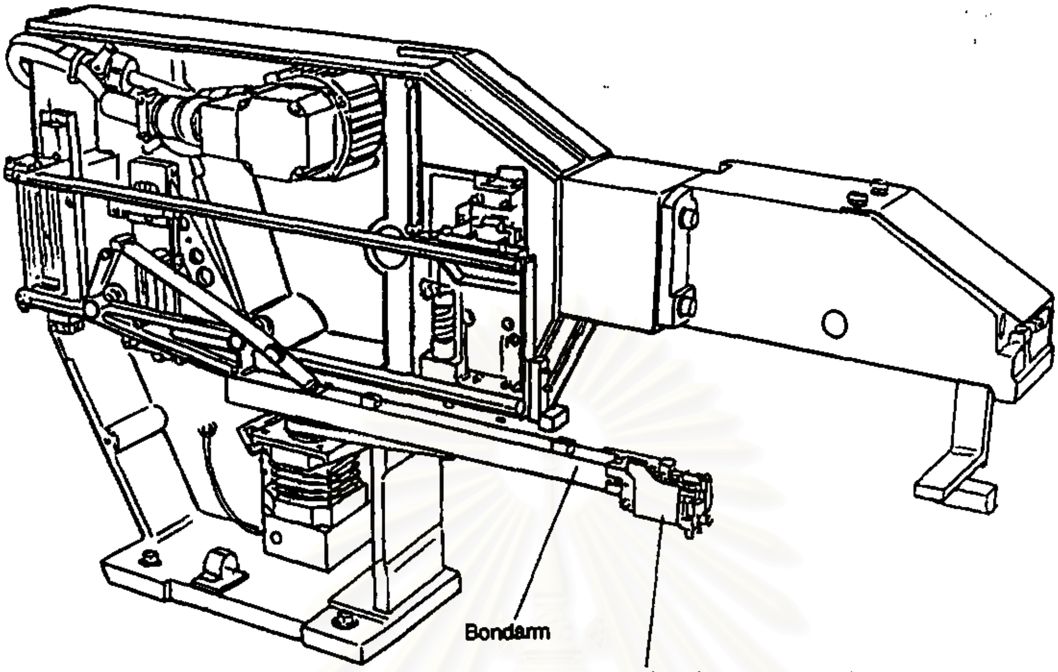

Figure 4.2: Bond arm and bond head of ESEC2007

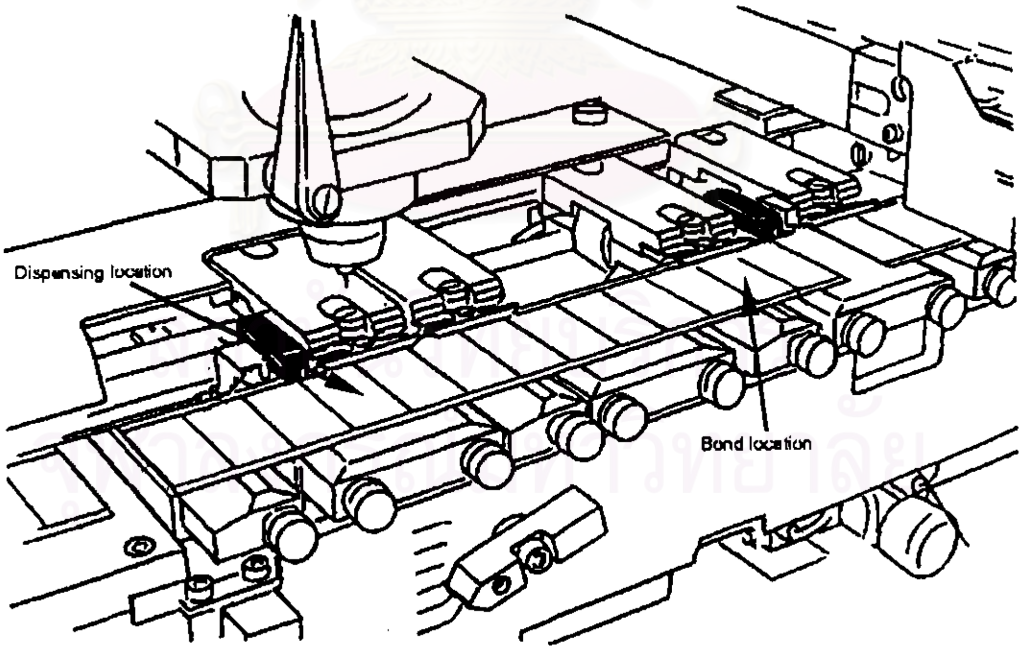

Figure 4.3: Indexing module.

The steps of die attach operation can be explained briefly by refer to the following figure.

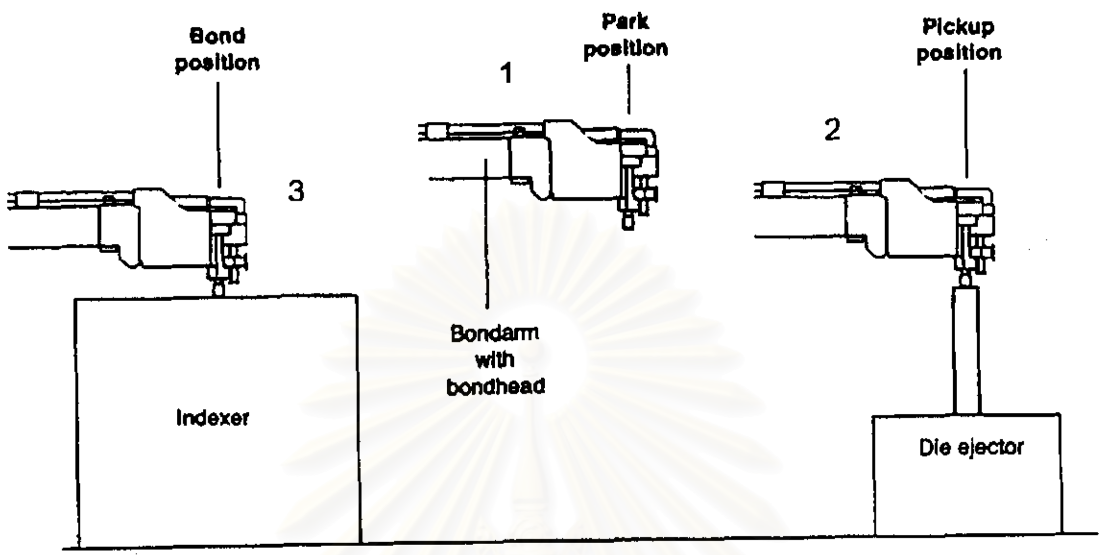

Figure 4.4: Steps of pickup and bond chip (die) onto leadframes.

The bond arm moves from park position (1) to pickup position (2) and picks a sawn chip from the wafer. At the same time, leadframes is placed onto the indexing module, and moved to dispensing location. The epoxy is applied on the leadframe at dispensing location. This epoxy is an adhesive between chip and leadframe. Leadframe is then moved onto bond location after applying the epoxy.

At bond location, the bond arm moves with a chip and places it onto the leadframes. After that, bond arm moves to park position and start for the next sequences.

After bonding step, the chip are attached onto die attach pad as shown in the below figure.

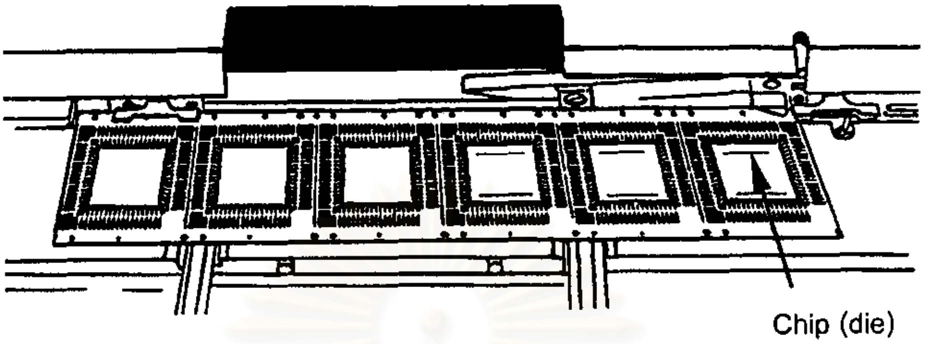

Figure 4.5: Die on leadframe after bonding.

The most errors of machine are in either leadframe indexing or chip bonding steps. The following cases are the examples of error that always occur during die attach machine operation.

1) Leadframe is damaged by indexer before reach bond location.
2) Leadframes can not be transferred to indexing module.
3) Chip drops during bond arm movement.
4) Chip is not attached onto the leadframe.

4.1.2 Host communications

The ESEC die attach machine model 2007 is designed for host communications. The communication interface between the machine and the host is in accordance with the SEMI Equipment Communication Standards SECS-I and SECS-II. The machine may be connected to a host computer via a standard computer RS-232C serial link. The machine provides a serial connector on the rear side so that this connector will be used for host interface. The following figure illustrates the location of serial communication connector on the machine.

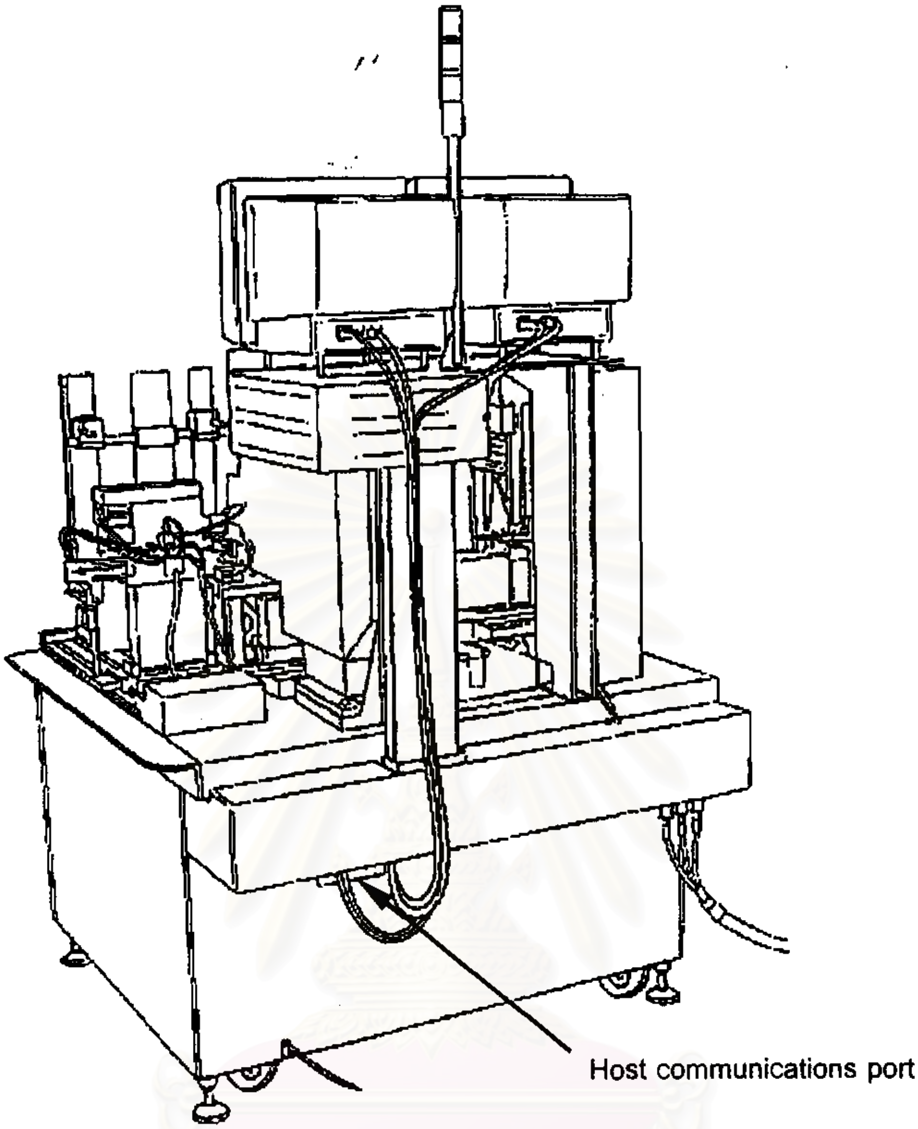

Figure 4.6: Rear side and serial communications connector location.

4.2 The standard of equipment communications

Semiconductor Equipment and Materials International (SEMI) is an international trade association. SEMI serves more than 1,600 member companies with programs, specifications, standards and guidelines to create a productive business environment for semiconductor manufacturing, such as chemicals and gasses, process equipment, factory communications protocols, facilities and safety.

SEMI's International Standard comprise the world's most comprehensive listing of standards specifications for needs, requirements and recommendations of vendors, users and industry consultants. Over 3,000 volunteers work together to develop the proposal of standards which are then submitted to among suppliers and customers for ballot. All approved specifications are published in eleven volumes which covering process chemicals, gasses, equipment automation on both hardware and software, materials, packaging, microlithography, facilities standards and safety guidelines.

According to the above information, standard of the equipment communications is also defined in SEMI's standards. For semiconductor equipment communications, the equipment suppliers have to design their products by refer to the SEMI's standard while the customers or users should also know the details of this standard. The development of equipment communications in the semiconductor industry has used the SEMI Equipment Communications Standard or SECS. SECS was chosen in order to meet the demands of managing the massive amounts of data generated. Moreover, it has been used for managing the expensive equipment in semiconductor manufacturing to expand its capability, The SECS is now being used generally in the other industries such as hard disk and head manufacturing, circuit board manufacturing, etc.

4.3 SEMI Equipment Communications Standard (SECS)

SECS is a co-ordinated pair of standards for the semiconductor industry that defines a serial communications interface between equipment and a host. The detail of SECS is defined in SEMI E4 (SECS-I) and SEMI E5 (SECS-II) of the SEMI standards. SECS–I defines a physical communications interface while SECS–II defines the content of the messages

4.3.1 SEMI E4 (SECS-I)

SECS–I is the general message transfer layer of the SECS standards. It defines the exchange of binary messages between a host and semiconductor processing equipment via RS-232, serial communications port. A host is an intelligent system of computer, which communicates with the equipment. Semiconductor processing equipment includes equipment intended for wafer manufacturing, wafer processing, assembly and packaging. This standard is established for independent manufacturers to produce equipment or host, which can be connected without requiring specific knowledge of each other. This standard includes the description of the physical connector, signal levels, data structure, data rate and message protocol.

4.3.1.1 Connector

As the previous mention, SECS-I is a standard for communications or exchange of binary messages between a host and semiconductor equipment via RS-232, serial communications port. Therefore, the RS-232 standard should be known.

The EIA RS-232 standard was established in 1960 to define the physical and logical interface between computer and communication equipment. The standard 25 pin connector type 'D' (DB-25) will be used. A connector on the equipment will be a female connector while a male connector will be mounted on the cable from the host. The maximum length of the cable is 50 feet.

4.3.1.1.1 Signal pins

Pins on the connector have functions as defined in the following table.

Pin	Description
1	*Positive Ground*
2	*Transmitted Data*
3	*Received Data*
4	Request To Send
5	Clear To Send
6	Data Set Ready
7	*Signal Ground*
8	Received Line Signal Detector
11	Select Standby
12	Secondary Receive Line Signal Detector
13	Secondary Clear To Send
14	Secondary Transmitted Data

15	Transmitter Signal Element Timing
16	Secondary Received data
17	Receiver Signal Element Timing
18	Test (+12 to +15 volts)
19	Secondary Request To Send
20	Data Terminal Ready
21	Signal Quality Detector
22	Ring / Calling Indicator
23	Data Signal Rate Selector
24	Transmitter Signal Element Timing
25	-12 to –15 volts

However, pin 1, 2, 3 and 7 are required for all equipment complying with SECS-I. Pin 18 and 25 are optional power supplies for driving external isolation circuits. If any other pins are used, it shall follow its function. The description of pins which are required for equipment communications per SECS-I standard shown as the followings:

Pin 1 (Protective Ground)

This pin will connect to body of the equipment as a ground.

Pin 2 (Transmitted Data Circuit, TD)

The data will be transferred from equipment through this pin.

Pin 3 (Received Data Circuit, RD)

The data will be sent from host to equipment through this pin.

Pin 7 (Signal Ground)

This line provides the reference signal of common ground for all data exchange circuits except protective ground.

4.3.1.1.2 Logic Levels

For the signal pins 2 and 3, a voltage less than –3 volts will be the logic 1; called "mark condition" and a voltage greater than +3 volts will be the logic 0 or "space condition". However, voltages must not exceed +/- 25 volts. The logic levels can be illustrated as the following figure.

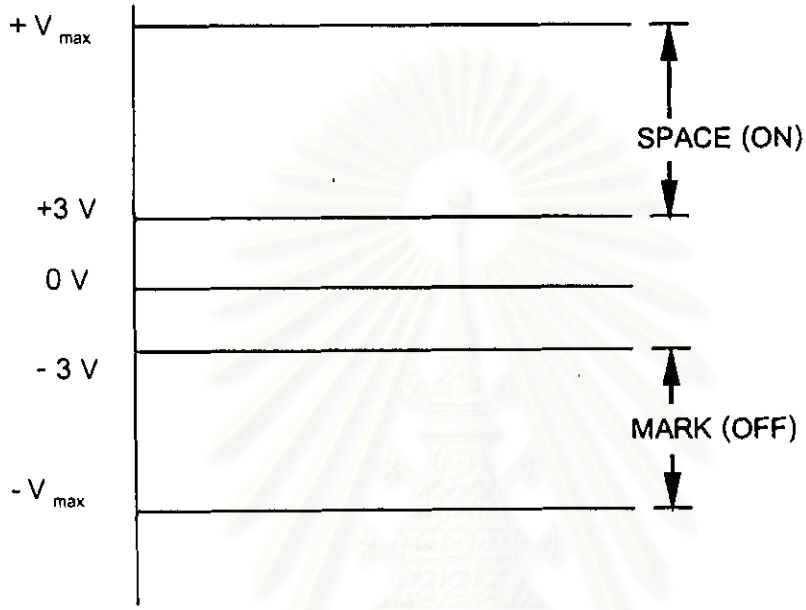

Figure 4.7: Logic levels of RS-232, serial communications port.

4.3.1.1.3 Data Rate

The data rates on signal pins shall be 9600, 4800, 2400, 1200 and 300 baud. However, it depends upon the capability of the equipment: The same data rate shall apply for data sent to and from the equipment.

4.3.1.2 Character Structure

The data transmission consists of 8-bit bytes sent serially with one start and one stop bit. All bit transmissions are the same duration. The 8 data bits are numbered from 1 to 8 in the order sent as shown in below figure. Bit one is the Least Significant Bit (LSB) and bit eight is the Most Significant Bit (MSB). The communication is bi-directional and asynchronous but the data flows in one direction at a time. Special characters and a handshake code define the direction of data transmission.

START BIT (0)	1	2	3	4	5	6	7	8	STOP BIT (1)

Figure 4.8: Character structure.

Data is sent in blocks of 254 bytes or less. Each block consists of a 10-byte header followed by data. For multi-block message, a message sent in more than one block, the maximum number of blocks that can be sent is 32,767 blocks.

4.3.1.3 Header structure

Data is sent in block with the maximum 254 bytes and the minimum 10 bytes. The first 10 bytes of every block are called "Header". The fixed format and structure of the header will be described. The following figure is the general header structure.

1	R	Upper device ID	
2		Lower device ID	
3	W	Upper message ID	STREAM
4		Lower message ID	FUNCTION
5	E	Upper block ID	
6		Lower block ID	
7		System	
8		System	
9		System	
10		System	

Figure 4.9: Header structure.

4.3.1.3.1 Reverse Bit. (R-Bit)

The reverse bit defines the direction of a message. The R-bit is set to "0" for message from host to equipment while set to "1" for message from the equipment to host.

4.3.1.3.2 Device ID

The device ID defines the source or destination of the message. It depends upon the value of the R-bit. If R-bit is set to "0", host to equipment, device ID will be destination. If R-bit is set to "1", equipment to host, device ID will be source. The host has no device ID so that device ID means equipment identification that wants to communication with.

4.3.1.3.3 Wait Bit. (W-Bit)

The wait bit is used to indicate that the sender of a primary message expects a reply. If W-bit is set to "0", a reply will not be expected. A value of one in this bit means that a reply is expected. For secondary message, the W-bit must be set to zero. For multi-block message, the W-bit must be the same in every block of the message.

4.3.1.3.4 Message ID

The message ID identifies the format and content of the message that is being sent. The primary message will have bit 1 of the lower message ID set to "1". Thus, it is defined as odd numbered message. The secondary message is defined as even numbered message and have bit 1 of the lower message ID set to "0". In SECS-II, byte three is defined as the "STREAM" and byte four of the header is known as the "FUNCTION".

4.3.1.3.5 End–Bit. (E–Bit)

The end bit is used to determine the block of a message. The last block of a message will have a value of E-bit set to one (1). A value of zero means that more blocks will be sent.

4.3.1.3.6 Block number

For multi-block message, more than one block is sent. Thus, the number of block shall be defined. The first block is given a block number of one, the block number for following block is incremented by one for each subsequent block unit the entire message are sent. In a single block message, the block number must have a value of zero or one. However, The maximum block number is 32,767.

4.3.1.3.7 System bytes

Byte 7 to byte 10 of the header of each message is system byte. It must satisfy the following requirement.

- They must be distinct from the most recently completed transaction.
- They must also be distinct from any system bytes of blocks that were not successfully sent since the last successful block send.
- The system bytes of the reply message must be the same as the system bytes of the corresponding primary message.
- The system bytes of all blocks of a multi-block message must be the same.

4.3.1.4 Block Transfer Protocol

The block transfer protocol is procedure used by the serial line to establish the direction of communication and provide the environment for passing message blocks. Most of the protocol is accomplished with a handshake of single bytes. The following are the definition of some words used in the block transfer protocol.

4.3.1.4.1 Timeout Parameters

The timeout are used to detect communication failures. It occurs when the measured time between two events exceeds a pre-determined limit. The time that required in one situation might be longer than another situation so that its values must be tuned to meet the application.

In the block transfer protocol, there are 2 situations requiring timeout values, Inter-Character Timeout (T1) and Protocol Timeout (T2).

1) Inter-character timeout (T1)

It limits the time between receipt of characters after the length bytes has been received and until the receipt of the second checksum byte.

2) Protocol timeout (T2)

It limits the time between the below situation.

1. Sending ENQ and receiving EOT.
2. Sending EOT and receiving length byte.
3. Sending the second checksum byte and receiving any character.

4.3.1.4.2 Handshake Bytes

There are four standard handshake codes used in the block transfer protocol. Its letter names and functions are shown as below.

Name	Code b8 b7 b6 ...... b1	Function
ENQ	0000 0101	Request to send
EOT	0000 0100	Ready to receive
ACK	0000 0110	Correct reception
NAK	0001 0101	Incorrect reception

4.3.1.4.3 Block lengths

The block length is the unsigned integer value that is sent after receipt of EOT. It defines the length of block that being sent. The length includes all bytes sent after the length byte excluding the 2 bytes of the checksum. However, the maximum block length is 254 bytes, and the minimum is 10 bytes.

4.3.1.4.4 Checksum

The checksum is calculated as the sum of the binary values of all bytes after the length byte and before the checksum. The checksum is sent as 16 bits in two bytes. The high orders eight bits of the checksum will be sent first, followed by the low order eight bits. The checksum is used by receiver to check the errors of transmission. The receiver performs the same checksum calculation on the received header and data, and then compare against the received checksum from sender.

4.3.1.4.5 Retry Limit (RTY)

If the time of transfer exceed the timeout parameters, the block transfer protocol will attempt to retry sending a block. The retry limits is the maximum number of the retry sending.

4.3.1.4.6 Master / Slave

When both ends of the line try to send message at the same time, a condition known as line contention. The protocol resolves contention by forcing one end of the line as the slave to postpone its transmission and enter the receive mode. Re-transmission of blocks is used to correct communication errors. The host is usually designated as the slave while the equipment is designated as the master. This is based upon the assumption that equipment is less able to store message than the host.

The operations of the block transfer protocol are shown and described in the following figure.

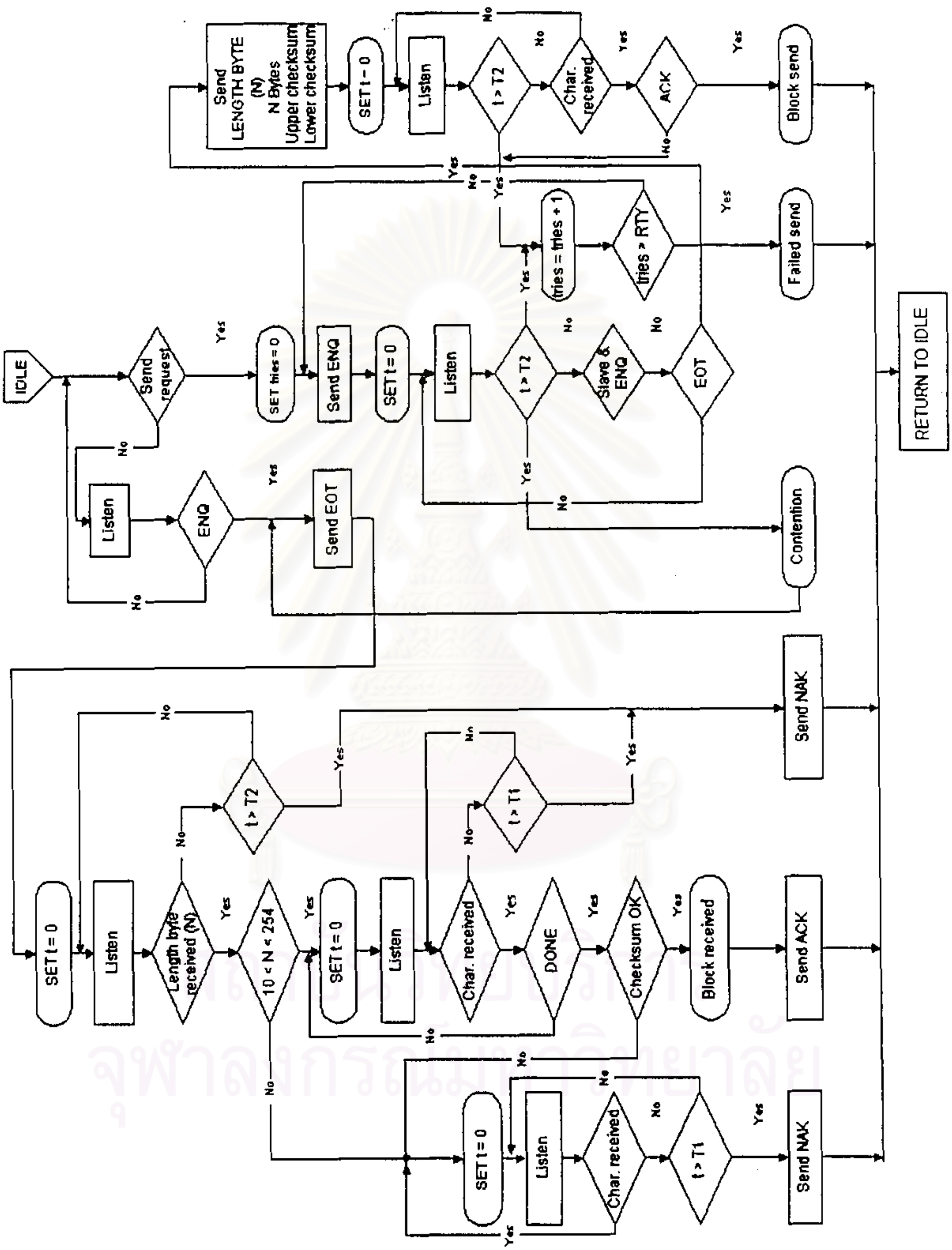

Figure 4.10: Block Transfer Protocol

The flow chart illustrates the operation of the five states of the protocol. It includes Idle, Receive, Send, Line control and Completion. The operation of each state is described as below.

1) Idle State

Both ends of the communication start in the idle state. There are two primary activities of the protocol can take the operation exits from the idle state. These are Send and Receive.

2) Send state

Once a message is to be sent, a sent state is established. The first byte sent is the block length (length of data in the block). After that, more bytes have been sent and then followed by the two bytes of the checksum. When the sender receives the ACK (Correct reception) before time T2, the block is deemed property sent. If the sender receives a non-ACK character before time T2, or no character sent within time T2, the line control state is established for possible retry.

3) Receive state

Once the other end of the communication has a message block to send, receive state is established. The first byte received is the block length byte. After that, the receiver counts and saves the following bytes including the checksum. The receiver computes the checksum bytes based upon the received data and then compares the two-checksum bytes together. The acceptable is both checksum bytes are the same.

4) Line control

The line control section establishes the transmission direction, resolves the contention and handles retries.

- Transmission direction

When an ENQ is received in the idle state, the line control responds with an EOT if the block transfer protocol is ready to receive. The block transfer protocol then goes to the receive state.

When a message is to be sent, and ENQ is sent to the receiver. If an EOT is received within the time limit T2, the block transfer protocol goes to the Send State.

- Line contention

If the slave receives an ENQ in response to the ENQ, the line contention has occurred. The slave will postpone the send of its block until it receives the block from the master. Thus, the slave prepares to receive the block and sends and EOT. The postponed block will be sent after the block transfer protocol returns to the Idle State.

When the master sends an ENQ, all characters can be ignored except an EOT. After a slave sends an ENQ, it can ignore all characters except an ENQ and EOT.

- Retries

When the time between sending ENQ and receiving EOT, or the time between sending the second byte of checksum and receiving any character exceeds the time limit T2, or a non-ACK character is received within time T2, the line control will go to retry sending. The retry counter will be increased and then re-send the message. If the retry count does not exceed the RTY parameter, the block transfer protocol will retry sending. If the retry count exceed the value of the RTY, a failed send has occurred.

5) Completion state

The completion state consists of receive completion and send completion.

- Receive completion

After the block is correctly received, an ACK character is sent to sender and the protocol is notified that a block has been received. However, receiver will send an NAK if it is the following situations.

1. T2 is exceeded while waiting for the length character.
2. T1 is exceeded between characters being received.
3. The length byte is invalid.
4. The received checksum does not agree with the computation.

The receiver may discard any received data after an NAK is sent. After that, the receiver returns to the Idle State immediately.

- Send completion

If an ACK is received within time T2, after the second checksum byte is sent, the message protocol is notified that it is successful sending.

4.3.1.5 Message protocol

A message is a complete unit of communication in one direction. It consists of the message data together with the following information from the header such as R-bit, W-bit, device ID, system bytes.

Before the message protocol explanation, some words, or parameters used will be described.

4.3.1.5.1 Message length

The maximum data length in a single block of a message is 244 bytes. (Maximum block length – header) The maximum number of blocks that can be sent in a multi-block message is 32,767. Therefore, the maximum data length allowed in a message is 244 x 32,767 bytes.

4.3.1.5.2 Transactions

A transaction is a primary message and a corresponding secondary message that is called the reply. A transaction is opened when primary message is ready to be sent. A transaction is closed when the last block of a primary message request no reply message or when the last block of the reply has been received.

4.3.1.5.3 Reply linking

When the primary message has a value of W–bit set to "1", a reply is expected. The expected block will have the complement of the R–bit, the same device ID, W-bit is "0", and have the same system bytes as those of the given primary message.

4.3.1.5.4 Reply timeout (T3)

The reply timeout limits the time that the message protocol is willing to wait after the last block of a primary message has been sent and before the arrival of the first block of the reply. If the first block of the reply does not arrive within the time T3, the expected block is removed from the list of expected block, and the transaction is aborted.

4.3.1.5.5 Inter-block timeout (T4)

The inter-block timeout limits the time interval between the successful receipt of a block in a multi-block message and the successful receipt of the subsequent block of the same message. If this time is exceeded, the message will be cancelled and the transaction is aborted. The inter-block timeout is used to measure the time between block arrivals in the multi-block message.

4.3.1.5.6 Routing error

When the equipment has received a block of data with device ID does not match its own device ID, it can assume that the block was sent in error.

When a block is sent successfully, all bytes in the header are combined to determine what to do with the block. The operation of the received message is shown in the following flow chart.

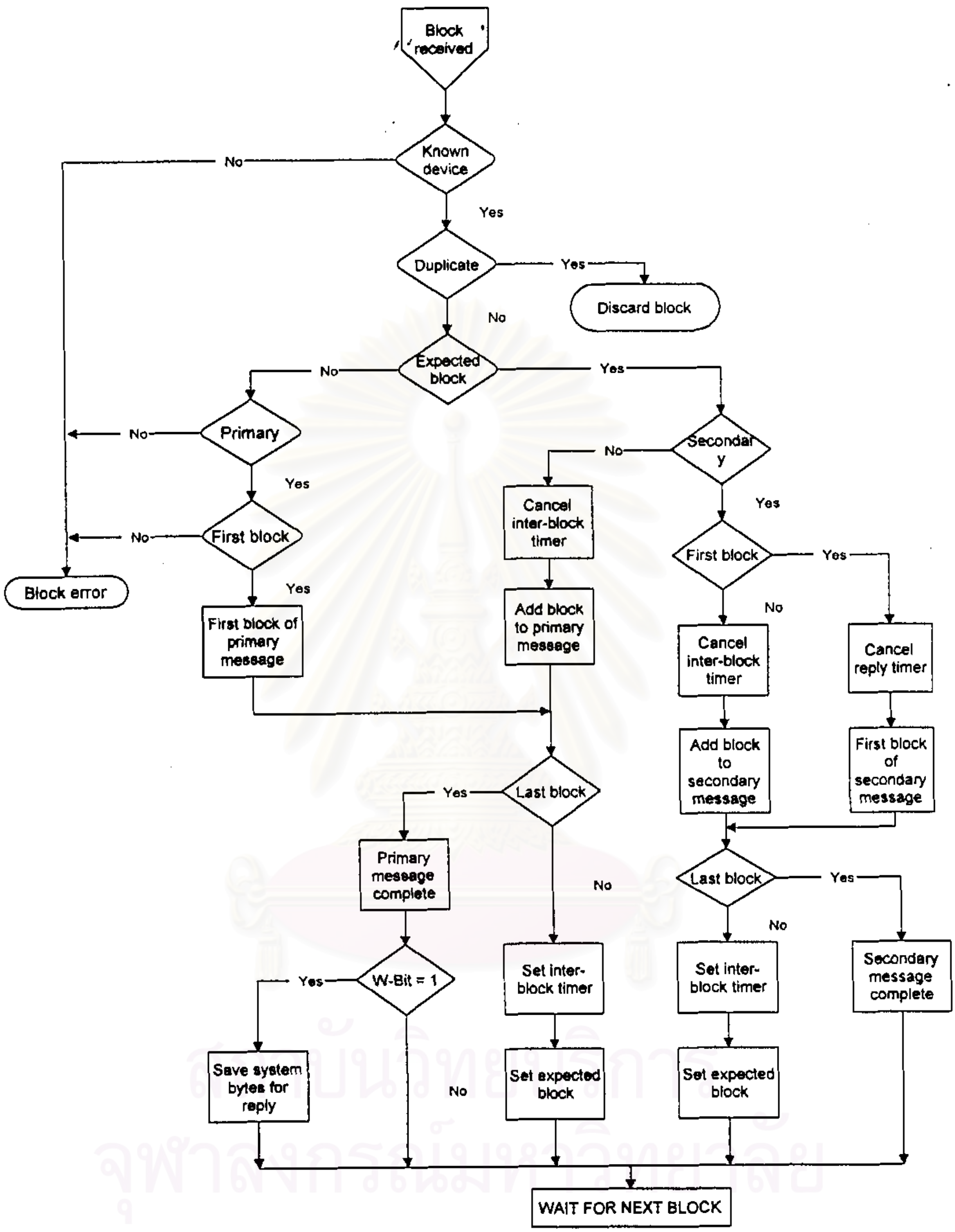

Figure 4.11: Message Receive Algorithm

The message protocol uses the concepts of an expected block. When a block is received, the first determination is whether the block is one of the expected blocks or not.

If the block is not one of the expected blocks, it must be a primary message otherwise the block has been sent in error. Thus, the block can be discarded. If the block is the first block of a primary message, and it is not last block of the message, an inter-block timeout (T4) is established and set.

If the block is one of the expected blocks, then it is either the first block of a reply message or it is part of an open message. If the block is the first block of a reply message, the reply timer for T3 is cancelled. If it is the last block of the given message, the message is complete

If the block is not the last block of the message, then the inter-block timer is reset, and the next expected block is set to have the same R-bit, device ID, W-bit, message ID, system bytes and the block number has one greater than the block that is just received.

4.3.1.6 Parameter setting

There are eight parameters of the block transfer protocol. The selection and setting are described as following.

1. **Baud rate (BAUD)**

   This parameter sets the serial line speed. It should be based on system performance.

2. **Device ID**

   It is identifier that is assigned to the equipment. The value of the device ID is determined by the particular system requirements and it's generally unique within one factory.

3. Inter-character timeout (T1)

The T1 detects the interruption between characters.

4. Protocol timeout (T2)

The T2 detects a lack of protocol response.

5. Reply timeout (T3).

The T3 detects a lack of reply message.

6. Inter-block timeout (T4).

The T4 detects an interruption in a multi-block message.

7. Retry limit (RTY).

Retry limit is the maximum number of send retries allowed.

8. Master / Slave (M/S)

The first 7 parameters must be adjustable by the user. The range and resolution of the parameters must be at least as shown in the following table.

Table 4.1: Parameters for equipment communications.

Parameter name	Typical value	Range	Resolution
Baud rate	9600	300 – 9600	-
Device ID	-	0 – 32767	1
Inter-character timeout	0.5 sec.	0.1 – 10 sec.	0.1 sec.
Protocol timeout	10 sec.	0.2 – 25 sec.	0.2 sec.
Reply timeout	45 sec.	1 – 120 sec.	1 sec.
Inter-block timeout	45 sec.	1 – 120 sec.	1 sec.
Retry limit	3	0 - 31	1

4.3.2 SEMI E5 (SECS-II)

SECS-II defines the details and method of conveying information between intelligent equipment and a host in the form of message. It intends to define message to such a level of detail that some host software may be constructed with only minimal knowledge of the equipment. Whilst the equipment requires minimal knowledge of the host. This standard is established to be fully compatible with SEMI Equipment Communication Standard E4 (SECS-I).

SECS-II provides form and meaning to message exchange between equipment and host using a message transfer protocol. The messages defined in the standard support the most typical activities required for Integrated Circuits (ICs) manufacturing. These messages are organised into categories of activities, called "**STREAMS**", which contain specific message, called "**FUNCTIONS**". This standard applies to equipment and hosts used in the manufacturing of semiconductor devices. Examples of the activities supported by the standard are transfer of control programs, material movement information, summarised test data, and alarms.

4.3.2.1 Streams and Functions

In SECS-II, messages are identified by a stream code and a function code. There are 127 stream codes defined in this standard by refer to 7 bits of the upper message ID in header. (Another bit is W-Bit). While all 8 bits of lower message ID in header are used for function codes so that 255 function codes are defined in the standard. Each combination of streams code and function code represents distinct message identification. The lists of SECS-II messages used in the ESEC die attach machine model 2007 and their structures are described in the **APPENDIX A**.

4.3.2.1.1 Streams

A stream is a category of message intended to support similar or related activities.

4.3.2.1.2 Functions

A function is a specifio message for a specific activity within a stream. All functions will follow a numbering convention corresponding to primary and secondary message pairs. As defined in SECS-I, all primary messages will be defined as odd numbered message so that it is given an odd numbered function code. Similarly, function code of the reply message is determined by adding one to the primary message function code. Function code "0" (zero) is reserved in all streams for aborting transactions.

4.3.2.1.3 Stream and Function Allocation

Some of stream and function combinations are reserved for this standard, while others are available for user definition. The stream and function codes reserved for this standard shown as follows:

Stream 0, Functions 0 - 255

Streams 1 - 63, Functions 0 - 63

Streams 64 - 127, Function 0

The stream and function codes available for user definition are as below:

Streams 1 - 63, Functions 64 - 255

Streams 64 - 127, Functions 1 - 255

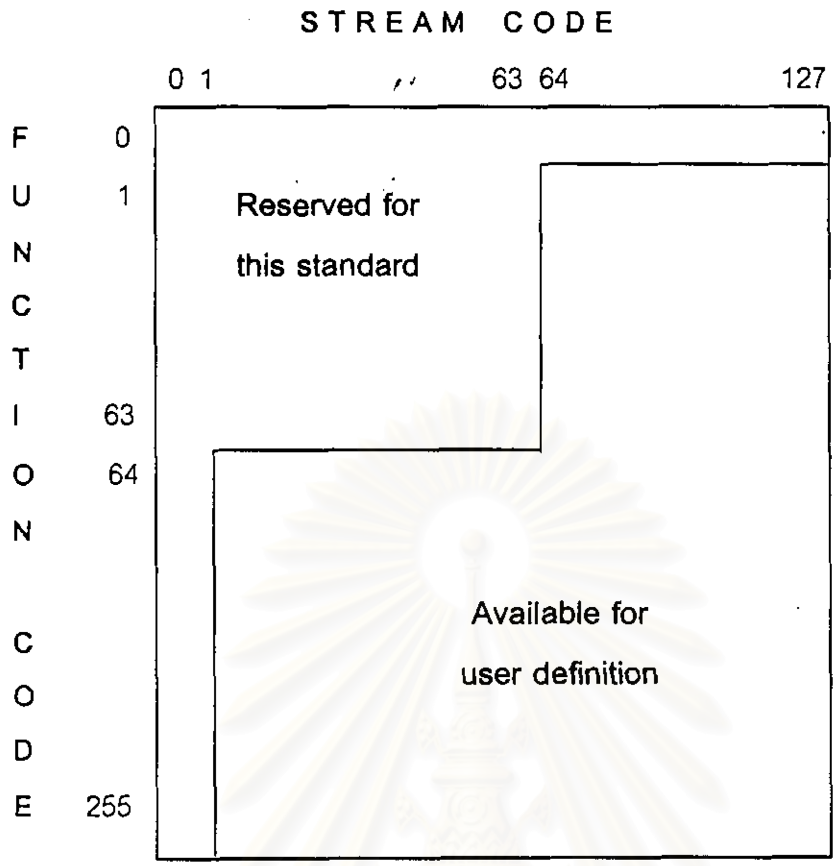

Figure 4.12: Streams and Function allocation.

4.3.2.2 Transaction protocol

In order to comply with SECS-II, the implementation must meet the minimum transaction requirements. The transaction forms the basis for all information exchanges in SECS-II. A transaction consists of either a primary message with no reply is requested, or a primary message which requests a reply together with its secondary message.

The transaction level requirements can be shown as the following items.

a) Respond to S1F1 with S1F2.

b) In case of any received message can not be processed by the equipment, the error message on stream 9 would be sent properly.

c) Send S9F9 to the host when a transaction timeout at equipment is detected.

d) When the function 0 is received as a reply to a primary message, the related transaction should be terminated.

4.3.2.3 Data structures

All information transmitted according to this standard will be formatted using two data structures, ITEMS and LISTS. These structures define the logical divisions of the message while message transfer protocol defines the physical divisions of the message.

4.3.2.3.1 Item

An item is an information packet, which has a length, and format defined by the first 2, 3, or 4 bytes of the item header (IH). The item header consists of the format byte and the length byte (s). Bit one and two of the format byte indicates the number of the following bytes, which refer to the length of the item. The item header can be shown as following figure.

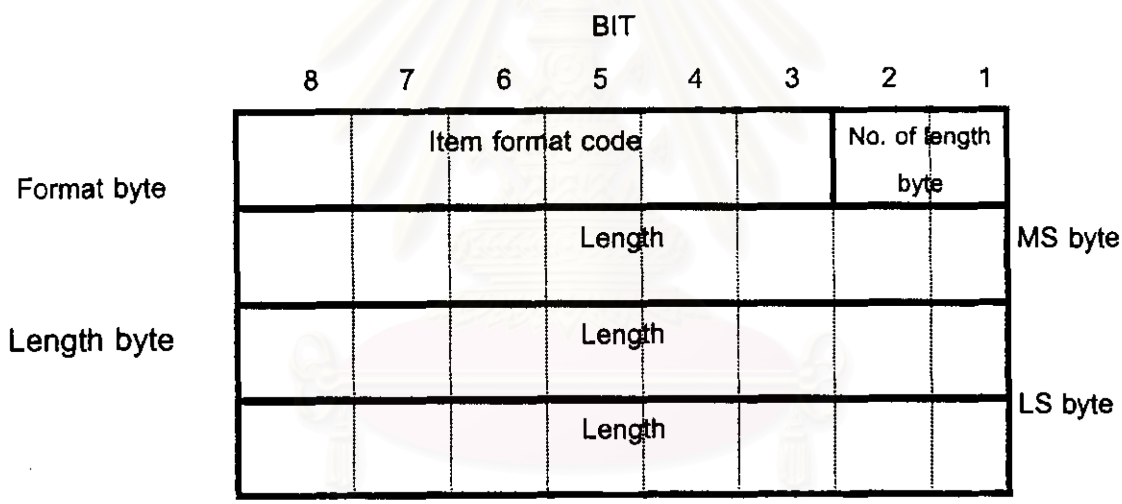

Figure 4.13: Data structure.

Number of length bytes can be 4 values, 0 (00), 1 (01), 2 (10), and 3 (11). The meaning of each can be described as below:

No. of length bytes	Meaning
0	Illegal or data format error
1	One binary length byte, maximum 255
2	Two binary length bytes, maximum 64K
3	Three binary length bytes, maximum 7.99M

The item length refers to the number of bytes following the item header called the **"item body"** (IB). It is the actual data of the item. The item length that is defined in item header refers to the item body only. It is not including the item header so that the actual number of bytes in the message is the item length plus 2, 3, or 4 bytes for the item header.

Bit three to eight of the format byte define the specific format of the item body. Hence, there are 64 possible formats, 000000 – 111111. The following table shows fifteen formats of the data.

	Format code		Meaning
	Binary	**Octal**	
Bit	**876543**		
	000000	00	LIST
	001000	01	Binary
	001001	11	Boolean
	010000	20	ASCII
	010001	21	JIS-8
	011000	30	8-byte integer (signed)
	011001	31	1-byte integer (signed)
	011010	32	2-byte integer (signed)
	011100	34	4-byte integer (signed)
	100000	40	8-byte floating point
	100100	44	4-byte floating point
	101000	50	8-byte integer (unsigned)
	101001	51	1-byte integer (unsigned)
	101010	52	2-byte integer (unsigned)
	101100	54	4-byte integer (unsigned)

Note: For integer, the most significant byte is sent first.

4.3.2.3.2 List

A list is an ordered set of elements, where an element can be either an item or a list. The list header has the same form as an item header with format code zero "0". The length bytes in list header refer to the number of elements in the list rather than the number of bytes. The list structure allows grouping items of related information, which may have different formats into a useful structure.

4.3.2.3.3 Example of data structures

a) An item contains two binary code of 10101010, 11001100.

Bit 8 7 6 5 4 3 2 1	
0 0 1 0 0 0 0 1	Item, binary, 1 length byte
0 0 0 0 0 0 1 0	2 bytes long
1 0 1 0 1 0 1 0	
1 1 0 0 1 1 0 0	

b) An item contains three ASCII characters, ABC.

Bit 8 7 6 5 4 3 2 1	
0 1 0 0 0 0 0 1	Item, ASCII, 1 length byte
0 0 0 0 0 0 1 1	3 bytes long
0 1 0 0 0 0 0 1	ASCII "A"
0 1 0 0 0 0 1 0	ASCII "B"
0 1 0 0 0 0 1 1	ASCII "C"

c) The machine temperature is measured by an equipment number 46. It found high temperature on the measured point T1. Hence, A message is sent from equipment to tell the host that the temperature at point T1 has exceeded a pre-set process limit. The message ID is stream 5, function 1 (Alarm report send), and the data consists of a list of three items. The first item is a code for the alarm set and the alarm category code. The second item is the equipment specific alarm number for this alarm, e.g. 17. The third item is a string of text indicates a description of the alarm, e.g. "T1 HIGH". This message dose not requests the reply message. The message including the message header is as follows:

Byte		
[Message Header]		
1	1000 0000	R=1,message from equipment to host
2	0010 1110	Device ID = 46
3	0000 0101	W = 0 (No reply), Stream 5
4	0000 0001	Function 1
5	1000 0000	E = 1 (Last block)
6	0000 0001	Block number 1
7	0000 0000	
8	0000 0000	System bytes = 0
9	0000 0000	
10	0000 0000	
[Data]		
11	0000 0001	List, 1 length byte
12	0000 0011	3 elements
13	0010 0001	1^{st} item, binary, 1 length byte
14	0000 0001	1 byte long
15	0000 0100	Alarm set, category 4
16	0110 0101	2^{nd} item, 1-byte integer, 1 length byte
17	0000 0001	1 byte long

18	0001 0001	Alarm 17
19	0100 0001	3rd item, ASCII, 1 byte length
20	0000 0111	7 characters
21	0101 0100	ASCII "T"
22	0011 0001	ASCII "1"
23	0010 0000	ASCII space
24	0100 1000	ASCII "H"
25	0100 1001	ASCII "I"
26	0100 0111	ASCII"G"
27	0100 1000	ASCII "H"

The message contains 1 byte length (not shown), 10 bytes of header, 17 bytes of data, and 2 bytes of checksum (not shown). The total byte for sending the above message is 30 bytes. At 9600-baud transmission rate, the message would be sent in 31 milliseconds.

4.3.2.4 Data items dictionary

This section defines the data items used in the SECS-II messages. These data items are stated in the structure of SECS-II messages. The name, format, and description of each data items are described in this dictionary. The APPENDIX B shows the data items dictionary of the SECS-II messages.

4.4 Communication interfaces on ESEC die attach machine model 2007

This section describes the supported state models of the ESEC die attach machine model 2007. It presents the state models for communication, equipment control and processing implemented on the equipment. These state models describe the behaviour of the equipment from a host perspective.

4.4.1 State model and methodology

The basic unit of a state model is the state. A state is a static set of conditions. These conditions might involve sensor readings, switch positions, etc. In order to clarify the state models, it is useful to distinguish between a state, an event and relationship of one to the other.

An event is dynamic rather than static set of conditions. It represents a change in conditions and the awareness of such a change. It might involve a sensor reading exceeding a limit, switch changing position, or a time exceeding.

A change to a new state, state transition, state must always be prompted by a change in conditions. In the other word, an event causes to a state transition in the equipment. In fact, there are many events which may occur on a piece of equipment, so it is important to classify events based on whether they can be detected and whether they are interesting.

A further narrowing of the definition of event is represented by the term "**Collection events**". It is an occurrence on the equipment, which is significant to the host. These collection events are reported to the host. The state models are intended to be limited to the level of detail in which the host is interested. Therefore, all state transitions will correspond to collection events.

Based on the communication interface manual of ESEC die bonder 2007, there are 3 state models that relate to the equipment i.e. communications state model, control state model, and equipment processing state model. The details of each state model are described in the following.

4.4.1.1 Communications state model

The communications state model defines the behaviour of the equipment in relation to the existence or absence of a communications link with the host. It relates to a logical connection between equipment and host rather than a physical connection.

4.4.1.1.1 Terminology

There are some terms that need to be clarified in order to eliminate the confusion with the same or similar terms.

1) Communication failure

A communication failure is detected through an unsuccessful attempt to send a message when equipment can not send a message after multiple attempts. This term occurs when the retry limit (RTY) is exceeded.

2) Connection transaction failure

This occurs when the equipment attempts to establish communications and is caused by:

1. a communication failure,
2. the failure to receive an S1, F14 reply within a transaction timeout limit, or
3. receipt of S1F14 with improper format or with establish communication acknowledge code (COMMACK) not set to zero.

3) Transaction timeout

According to the block transfer protocol, an ACK is sent after receiving the correct primary message. If a reply is expected, a transaction timeout period will be activated immediately. A transaction timeout period begins after the successful transmission of a complete primary message. The transaction timeout in a SECS-I indicates that the equipment did not correctly receive the first block of the reply message within the defined timeout limit.

4) Communication link

A communication link is the first successful completion of any one S1, F13/F14 transaction with an acknowledgement of "accept". It is logical link rather than physical link.

5) Establish communications timeout

Establish communications timeout is the equipment constant that defines the delay between attempts to send S1F13. This value is used to initialise the CommDelay timer.

6) CommDelay timer

CommDelay timer is an internal timer that is used to measure the interval between attempts to send S1F13. This interval is equal to the value in the "Establish communications timeout".

4.4.1.1.2 Communications states

There are 2 major states of SECS communications in the ESEC die bonder model 2007, DISABLED and ENABED. The operator is able to change the communications state selection via function keys on the front panel. The following figure illustrates the main window and function keys position to enable and disable SECS communications.

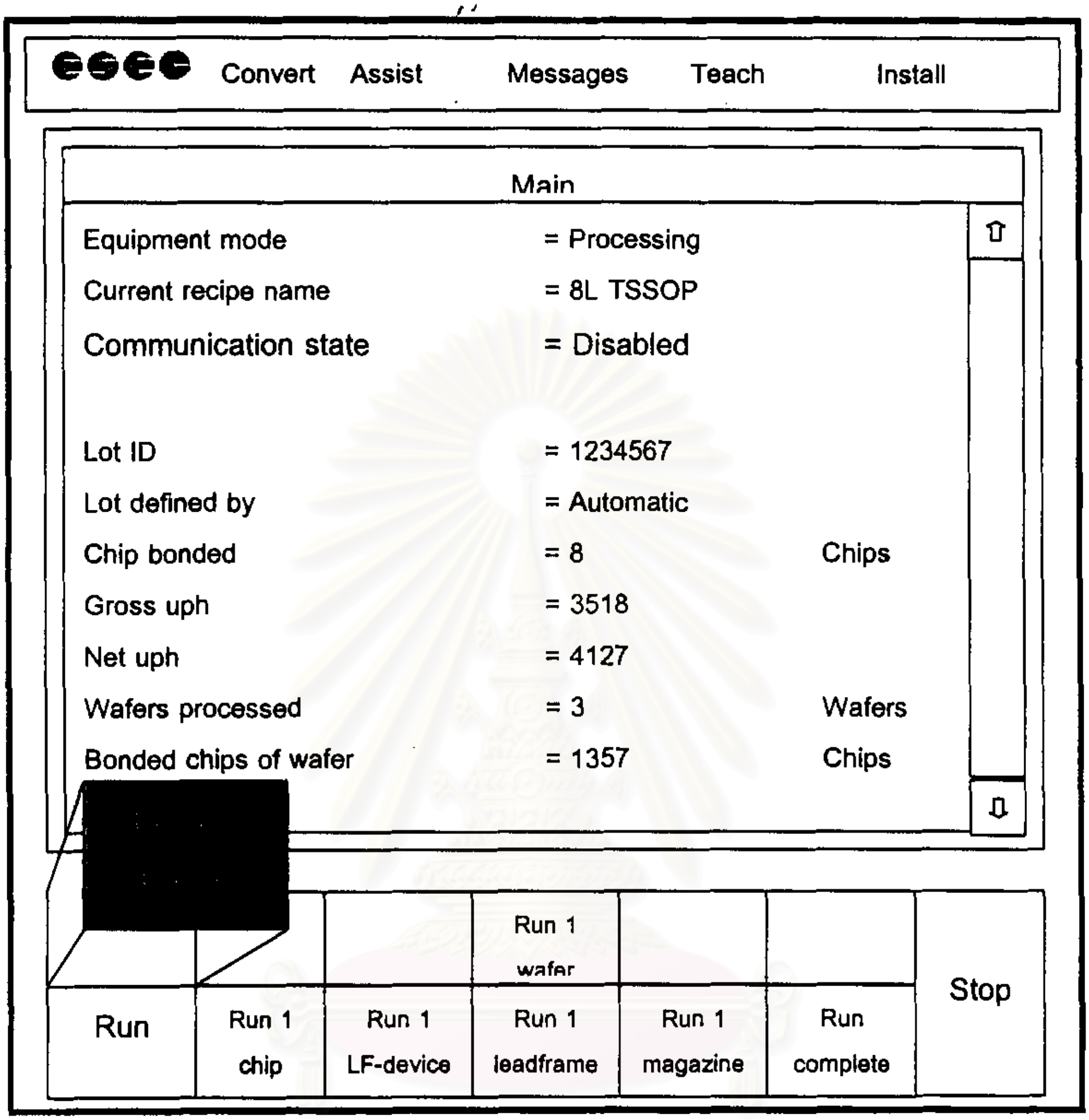

Figure 4.14: Main window and function keys for SECS communications.

The ENABLED state has two substates, NOT COMMUNICATING and COMMUNICATING. The current communications state is displayed in the equipment main window continuously. Figure 4.15 shows the relationship between the super-states and substates of the communications state model.

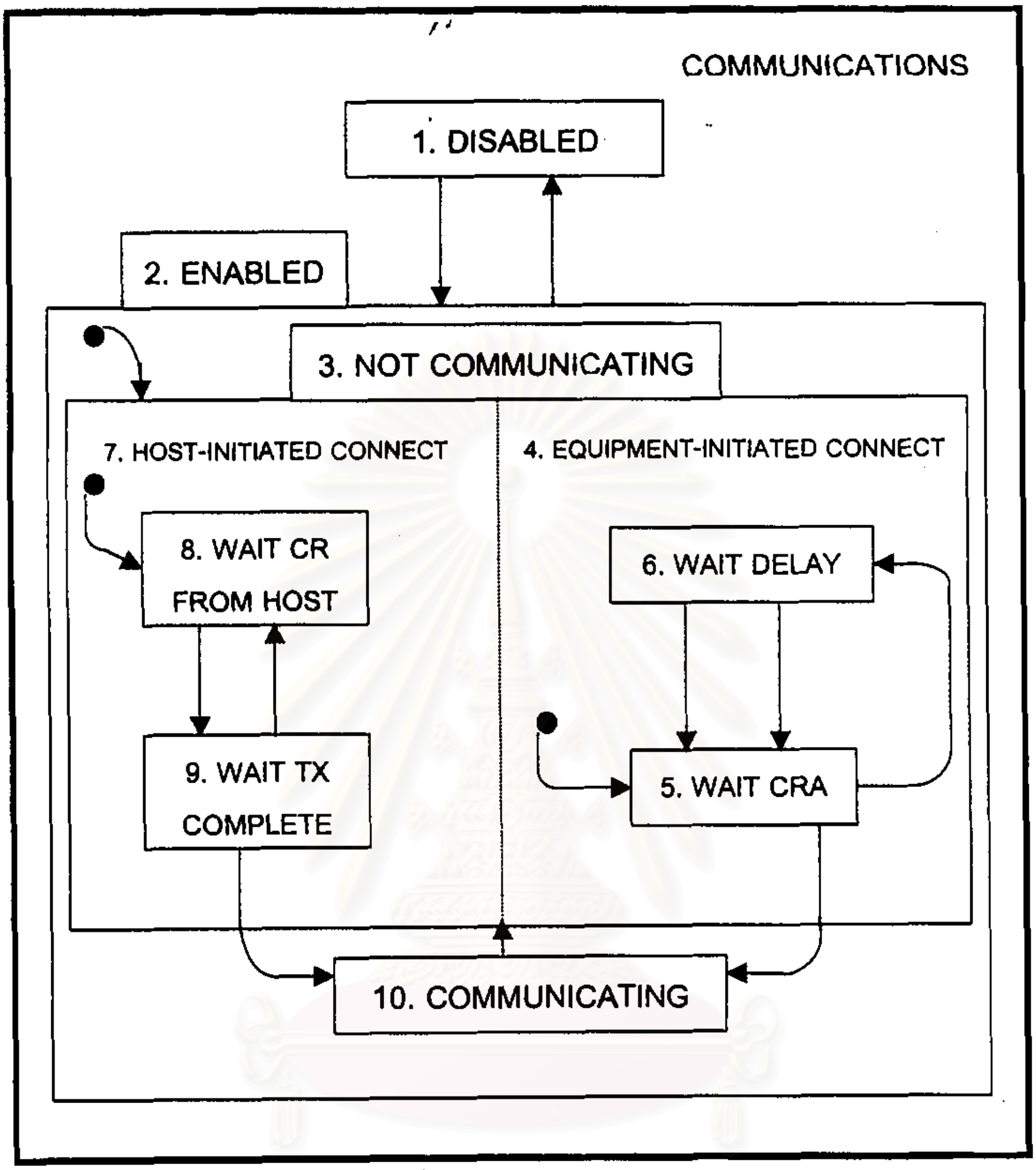

Figure 4.15: Communications state diagram.

The states of the communications state model are defines as follows:

1. DISABLED

In this state, SECS-II communications with a host computer is non-existent. If the operator switches from ENABLED to DISABLED, all SECS-II communications will cease immediately. There is no message transferred between the equipment and host in the DISABLED state.

2. ENABLED

There are 2 substates for ENABLED state, COMMUNICATING and NOT COMMUNICATING. Whenever communications are enabled, the substate of NOT COMMUNICATING is active until communications are formally established.

When the operator switches from the DISABLED state to the ENABLED state, no collection event occurs because no messages can be sent between the equipment and the host until communications have been established.

3. ENABLED / NOT COMMUNICATING

There are no message sent from the equipment while this substate is active except S1F13, S1F14 and S9Fx. The equipment discards any messages that are received from the host other than S1F13 or S1F14. The equipment also periodically attempts to establish communication with a host computer by issuing an S1F13, until communications are successfully established.

The NOT COMMUNICATING state has two substates, HOST-INITIATED CONNECT and EQUIPMENT-INITIATED CONNECT. Both of the substates are active whenever the equipment is in the state of NOT COMMUNICATING. These two substates illustrate the behaviour of the equipment in the event that both the equipment and the host attempt to establish communications during the same period of time.

4. NOT COMMUNICATING / EQUIPMENT-INITIATED CONNECT

This state has two substates, WAIT CRA and WAIT DELAY. Whenever EQUIPMENT-INITIATED CONNECT first become active, the state of WAIT DELAY is entered and the CommDelay timer is initialised to begin timing by refer to the value that stated in the "Establish communications timeout". After the state WAIT DELAY, and the time is initialised, the equipment attempts to send an S1F13 and transit into the state WAIT CRA. If the equipment does not receive S1F14 and the time remaining in the interval between attempts to send is zero, the CommDelay timer is expired. Thus, it enters to WAIT DELAY again, and triggers a new attempt to send S1F13 with transit into the state of WAIT CRA.

Note that the attempt to send S1F13 is made only upon transit into state WAIT CRA while the timer is initialised only upon transit into state WAIT DELAY.

5. NOT COMMUNICATING / EQUIPMENT-INITIATED CONNECT / WAIT CRA

An Establish Communications Request (S1F13) has been sent by the equipment. The equipment waits for the host to acknowledge the request by sending Establish Communications Request Acknowledge (S1F14) with COMMACK = 0. If S1F14 is received completely, communications have been successfully established.

6. NOT COMMUNICATING / EQUIPMENT-INITIATED CONNECT / WAIT DELAY

Whenever a connection transaction failure has occurred, the CommDelay timer has been initialised. Therefore, the equipment waits for the timer to expire.

7. NOT COMMUNICATING / HOST-INITIATED CONNECT

HOST-INITIATED CONNECT has two substates, WAIT CR FROM HOST and WAIT TX COMPLETE. When HOST-INITIATED CONNECT first becomes active, a transition to WAIT CR FROM HOST occurs and the equipment begins waiting for an S1F13 from the host.

8. NOT COMMUNICATING/HOST-INITIATED CONNECT/WAIT CR FROM HOST

In this state, the equipment waits for an S1F13 from the host. If an S1F13 is received, the equipment attempts to send an S1F14 with COMMACK = 0.

9. NOT COMMUNICATING/HOST-INITIATED CONNECT/WAIT TX COMPLETE

This state becomes active after an attempt to send an S1F14 begins. It remains active until the completion of the S1F14 transmission. If the transmission of S1F14 is successful, communications have been established.

10. ENABLED / COMMUNICATING

This state indicates that the communications have been established. The equipment may receive any message from the host. When the equipment is COMMUNICATING, SECS communications with a host computer is maintained. The COMMUNICATING state remains active until communications are disabled or a communication failure occurs. In the event of communication failure, the equipment returns to the NOT COMMUNICATING substate and attempts to re-establish communications with the host. The details of establish communications will be described in the following section.

After the communication is established, the communication state on main window will be changed to be COMMUNICATING, and a function key of "ON LINE" command appears as shown in the figure 4.16.

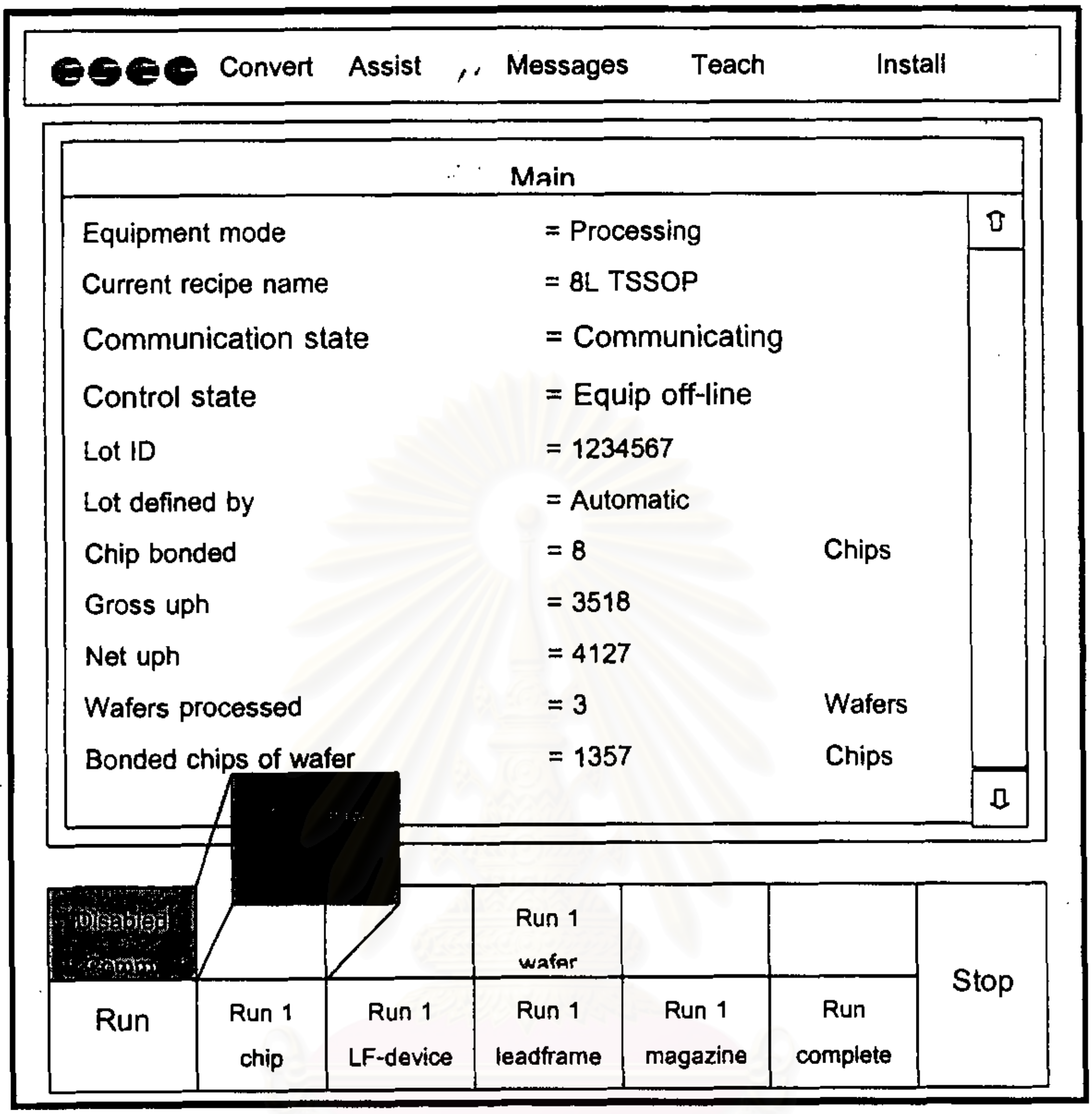

Figure 4.16: Main window and function keys after successful establishment of communications.

The following table shows the description of the events triggering state transitions and the actions taken.

Table 4.2: Communications state transition table.

#	Current State	Trigger	New State	Action	Comment
1	(Entry to Communications)	System initialization	System default	None.	The system default may be set to DISABLED or ENABLED.
2	DISABLED	Operator switches from DISABLED to ENABLED.	ENABLED	None.	SECS-II communications are enabled.
3	EBABLED	Operator switches from DISABLED to ENABLED.	DISABLED	None.	SECS-II communications are prohibited.
4	(Entry to ENABLED)	Any entry to ENABLED state.	NOT COM-MUNICATING	None.	May enter from system initialization to ENABLED or through operator switch to ENABLED.
5	(Ebtry to EQUIPMENT-INTIATED CONNECT)	Aby entry to NOT COMMUNICATING.	WAIT CRA	Intialize communicaions. Set CommDelay time "expired". Send S1,F13.	Begin to attempt establish communications.
6	WAIT CRA	Connection transaction failure.	WAIT DELAY	Intialize CommDelay time.	Wait for time to expire.
7	WAIT DELAY	CommDelay timer expired.	WAIT CRA	Send S1,F13.	Wait for S1,F14. May receive S1,F13 from Host.
8	WAIT DELAY	Received a message other than S1,F13.	WAIT CRA	Discard message. No Reply. Set CommDelay timer "expired". Send S1,F13.	Indicates opportunity to establish communications.
9	WAIT CRA	Received expected S1,F14 with COMMACK = 0.	COMMUNICATING	None.	Communications established.
10	Entry to HOSTINTIATED CONNECT)	Aby entry to NOT COMMUNICATING.	WAIT CR FROM HOST	None.	Wait for S1,F13 from Host.
11	WAIT CR FROM HOST	Received S1, F13	WAIT TX COMPLETE	Send S1,F14 with COMMACK = 0.	Host seeks to establish communications.
12	WAIT TX COMPLETE	S1,F14 transmission failed.	WAIT CR FROM HOST	None.	Wait for S1,F13 from Host
13	WAIT TX COMPLETE	S1,F14 transmission completed successfully.	COMMUNICATING	None.	Communications are established.
14	COMMUNICATING	Communication failure.	NOT COMMUNICATING		

4.4.1.1.3 Control State model

The control state model defines that level of co-operation between the host and equipment. It specifies how the operator interacts at different levels of host control.

There are three basic levels of control in the control state model i.e. REMOTE, LOCAL, and OFF-LINE. The highest level is remote level that the host may control the equipment to the full extent possible. The local level is middle level, which allows host to fully access to the equipment's information, but places some limitations on how the host can affect equipment operation. In the lowest level, off-line, the equipment does not allows any host to control them however; some information of the equipment may be accessed with very limitation.

The control state model and communications state model does not interact directly so that action of one model does not cause to change in behaviour of the other. However, when the communication is in NOT COMMUNICATING state the messages are not transmitted.

The figure 4.17 illustrates the control substates and state transitions in the control state model.

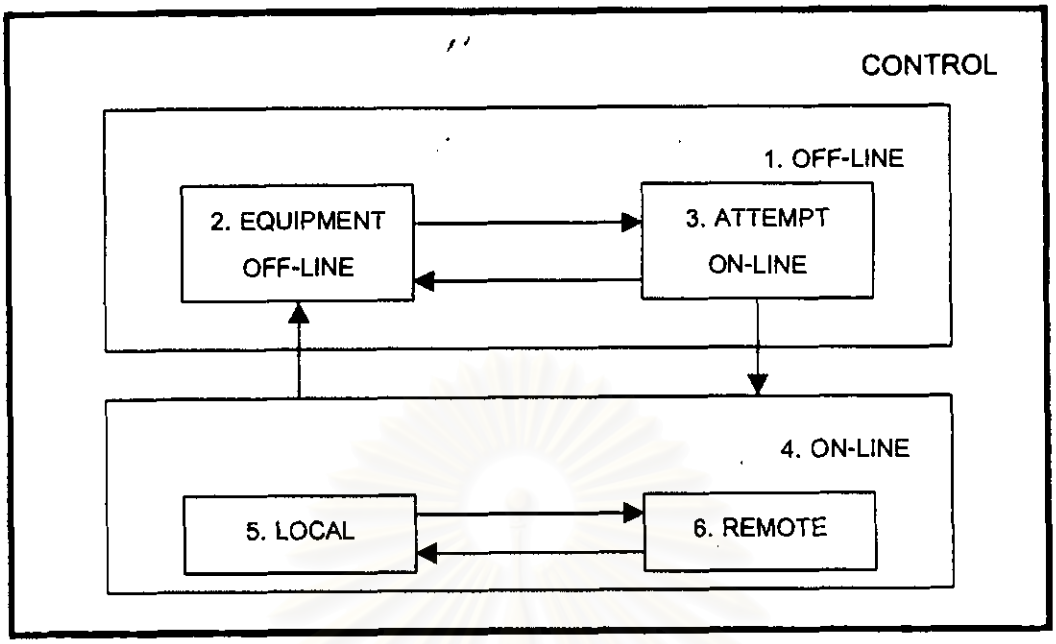

Figure 4.17: Control state diagram.

1. OFF-LINE

In this state, operation of the equipment is performed by the operator. While the equipment is OFF-LINE, messages transfer is possible. However, the use of messaging for automation purpose is restricted. Only the messages of establish communications such as S1F13 can be responded by the equipment when the OFF-LINE state is active. The other primary messages from the host will be responded with Abort transaction message (SxF0)

While the OFF-LINE state is active, the equipment does not send any primary messages other than S1F13, S9Fx, and S1F1 (Are you there request). If the equipment receives a reply message from the host other than S1F14 or S1F2, this message will be discarded. The OFF-LINE state has two substates, EQUIPMENT OFF-LINE and ATTEMPT ON-LINE.

2. OFF-LINE / EQUIPMENT OFF-LINE

In this state, the equipment waits for the instructions to attempt to go ON-LINE state. The operator has to press ON-LINE key to activate the ATTEMPT ON-LINE state.

3. OFF-LINE / ATTEMPT ON-LINE

After the operator has pressed ON-LINE key, the control state transits to ATTEMPT ON-LINE substate. The equipment attempts to send an S1F1 to the host. This indicates that the equipment attempts to go to ON-LINE state.

4. ON-LINE

In the ON-LINE state, the SECS-II messages from the host are exchanged and acted upon. Moreover, in case of system error the stream 9 messages will be used instead of SxF0 (Abort transaction message). There are two substates of ON-LINE state i.e. LOCAL and REMOTE

5. ON-LINE / LOCAL

While the LOCAL state is active, operation of the equipment is implemented by direct action of an operator. The equipment acts based on the command from the local operator console.

The host has following capabilities and restrictions when the LOCAL state is active.

- The host can not use any remote command and is prohibited from modifying any equipment constants.
- The host is able to configure automatic data reporting including alarms, event reporting, and trace data reporting. It will be sent to the host at the appropriate times.
- The host is able to inquire for data from the equipment, including status variable data, equipment constants, event reports, and alarms.

6. ON-LINE / REMOTE

While the REMOTE state is active, the host may operate the equipment with full extent available through the communications interface. There are no any restrictions on the capabilities when REMOTE state is active.

However, the operator may be restricted in specific capabilities. The equipment may be set up to allow the operator to perform necessary functions without contention with the host. These restrictions include:

- change equipment constants on both process related and non-process related.
- initiate process program download.
- select process program.
- start process programs.
- pause / resume process program.
- operator assist.
- material movement to / from equipment.
- equipment-specific commands.

Table 4.3 contains a full description of the control state transition that is depicted in figure 4.17

Table 4.3: Control state transition table.

#	Current State	Trigger	New State	Action	Comment
1	(Undefined)	Entry into CONTROL state (system initialization).	CONTROL (Substate conditional on configuration).	None.	Equipment may be configured to default to ON-LINE or OFF-LINE[1].
2	(Undefined)	Entry into OFF-LINE state.	OFF-LINE (Substate conditional on configuration).	None.	Equipment may be configured to any substate of OFF-LINE.
3	EQUIPMENT OFF-LINE	Operator actuates ON-LINE switch.	ATTEMPT ON-LINE	None.	Note that an S1,F1 is sent whenever ATTEMPT ON-LINE is activated.
4	ATTEMPT ON-LINE	S1,F1 transaction terminates without receipt of S1,F2.	EQUIPMENT OFF-LINE	None.	This may be due to a communication failure[2], reply timeout, or receipt of S1,F0.
5	ATTEMPT ON-LINE	Equipment receives expected S1,F2 message from the host.	ON-LINE	None.	Host is notified of transition to ON-LINE state at transition 7.
6	ON-LINE	Operator actuates ON-LINE switch.	EQUIPMENT OFF-LINE	None.	"Equipment OFF-LINE" event occurs[3]. Event reply will be discarded while OFF-LINE is active.
7	(Undefined)	Entry into ON-LINE state.	ON-LINE (Substate conditional on REMOTE/ LOCAL switch setting).	None.	"Control State LOCAL" or "Control State REMOTE" event occurs. Event reported based on actual ON-LINE substate activated.
8	LOCAL	operator actuates REMOTE switch.	REMOTE	None.	"Control State REMOTE" event occurs.
9	REMOTE	Operator actuates LOCAL switch.	LOCAL	None.	"Control State LOCAL" event occurs.

4.4.1.1.4 Equipment processing state model

The equipment processing state model defines the behaviour of the equipment in the performance of its intended functions. The figure 4.7 is processing state diagram that defines the state and state transitions of the equipment processing state model. The equipment generates collection events for each processing state transition.

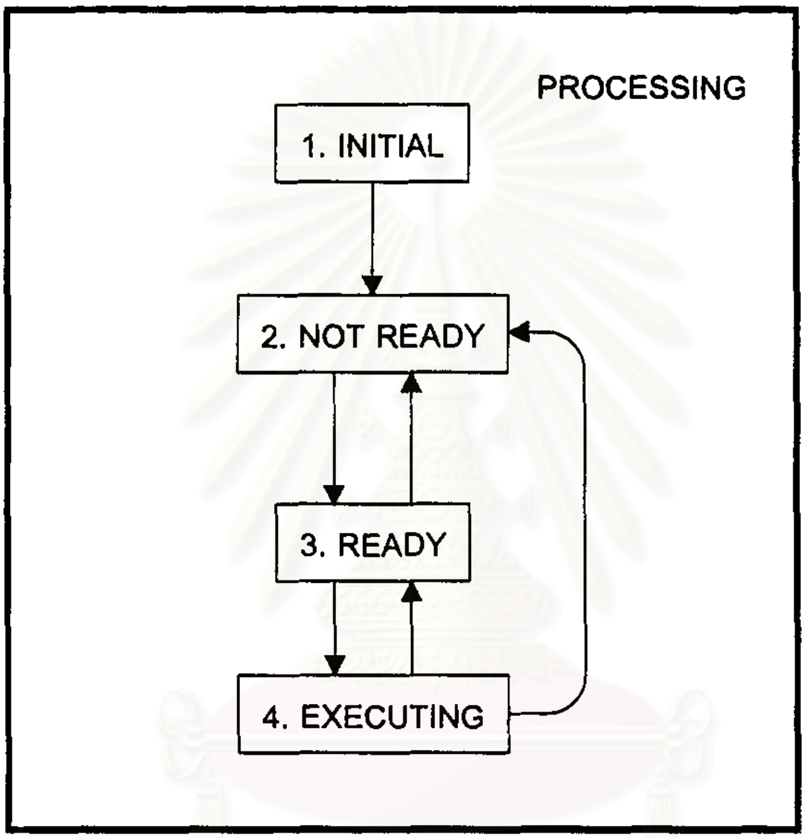

Figure 4.18: Processing state diagram

The details of each processing state are described in the following.

1. INITIAL

The initialisation state is not an actual processing state. It is shown to indicate that the NOT READY processing state is not entered after completion of equipment system initialisation.

2. NOT READY

When the equipment is in this state, means that it is not capable to process product automatically. The equipment requires some external assistance. In this state all activities can take place except the execution of a process program.

3. READY

In this state the equipment is capable to process product automatically and it waits for a start command.

4. EXECUTING

When the operator press "Start" key, the equipment enters to this state. In this state the equipment is automatically processing material and can continue to do without any external intervention.

The description of equipment processing state transition and its actions can be described in the table 4.4

Table 4.4: Processing state transition.

#	Current State	Trigger	New State	Action	Comment
1	INIT	Equipment initialization complete.	NOT READY	None.	None.
2	NOT READY	All setup activity has completed and the equipment is ready to receive a START command.	READY	None.	None.
3	READY	The operator selected a setup function or an abnormal situation requiring operator interaction has been detected in an elementary operation.	NOT READY	None.	None.
4	READY	The equipment has received a START command from the host or the operator console.	EXECUTING	Equipment begins to process automatically material.	None.
5	EXECUTING	The processing task has been completed or the equipment has received a STOP command from the host or the operator console.	READY	Equipment stop automatic material processing letting the equipment in a safe state.	None.
6	EXECUTING	The equipment detected an abnormal situation requiring operator interaction.	NOT READY	Equipment stop automatic material processing letting the equipment in a safe state.	None.

4.4.2 Equipment capabilities and scenarios

This section provides a detailed description of the communication capabilities defined for the ESEC die attach machine model 2007. It also provides the scenarios for their use.

Capabilities are operations performed by the equipment. These operations are initiated through the communications interface using SECS-II messages.

A scenario is a group of SECS-II messages that are arranged in a sequence to perform a capability.

The following are equipment capabilities of ESEC die attach machine model 2007 that relate to SECS communications.

1) Establish communications.
2) Control.
3) Alarm management and exception reporting.
4) Remote control.
5) Equipment constant.
6) Process program management.
7) Material movement.
8) Data collection.
9) Wafer mapping.

Based on the above list, only some equipment capabilities relate to the **"ERROR DETECTION SYSTEM"**. Those capabilities include

1) Establish communications.
2) Control
3) Data collection.
4) Alarm management and exception reporting.
5) Material movement.

Any way, there are many technical words and terms used in this section. Therefore, it needs to be clarified before explanation on detailed description of each capability.

4.4.2.1 Definitions

Alarm – The abnormal situation on the equipment that may endanger people, equipment, or materials being processed. Such abnormal situations are defined by ESEC SA based on physical safety limitations.

Buffer – A storage area defined at an equipment port.

Collection event – An event on the equipment that is considered significant to the host.

Collection Event ID (CEID) – A unique identifier of a collection event.

COMMACK – Acknowledge code returned in the Establish Communications Request Acknowledge message.

Establish communication timeout – An equipment constants that are used to initialise the interval between attempts to send an S1F13.

Event – A detectable occurrence that is significant to the equipment.

Port – A point or area on the equipment at which a change in equipment ownership of material may occur.

Status variable value (SV) – A data item that contains the value of a status variable.

Status Variable ID (SVID) – A unique identifier of a status variable.

4.4.2.2 Establish communications

This capability refers to the communications state model. The Establish Communications capability provides a means of formally establishing communications, in which relate to system initialisation or any loss of communications between the equipment and host.

Communications between the host and equipment are formally established through use of the Establish Communications Request (S1F13) and Establish Communications Request Acknowledge (S1F14) transaction.

The S1F13 / F14 transaction provide a means for equipment to notify the host, or the host to notify the equipment that there is a period of inability to communicate between them.

Equipment considers communications as formally established whenever, EITHER of the following conditions have been satisfied.

1) An Establish Communications Request (S1F13) has been sent to the host and an Establish Communications Request Acknowledge (S1F14) with an acknowledge code of accept has been received within the transaction timeout period.
2) An Establish Communications Request (S1F13) has been sent from the host and an Establish Communications Request Acknowledge (S1F14) with acknowledge code of accept has been successfully sent to the host.

Both conditions can be illustrated by the following figure.

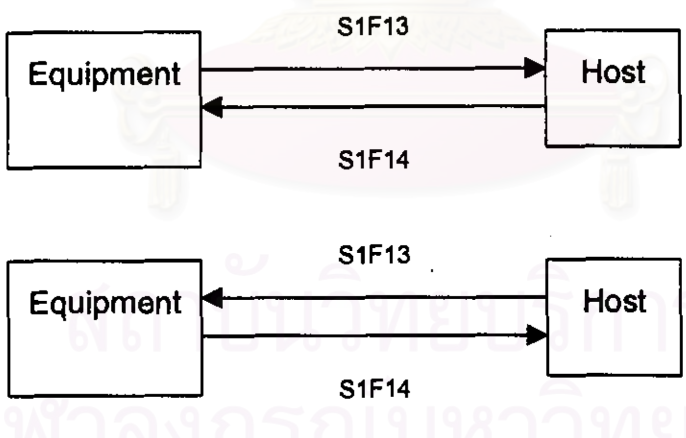

Figure 4.19: Establish communications messages transaction between the equipment and host.

The satisfaction of either of above conditions will result in a transition to the communications state as described in the communications state model.

It is possible that the equipment may be awaiting an S1F14 from the host while an S1F13 is sent from the host. When this situation occurs, both equipment and host have an open S1F13 / F14 transaction. Either of these two transactions may be the first to complete successfully since communications are established on successful completion of any S1F13 / F14 transaction. Hence, the communications state transits from NOT COMMUNICATING state to COMMUNICATING state with a transaction is closed.

However, another S1F13 / F14 is still open so that an Establish Communications Request Acknowledge (S1F14) is required to close this transaction as shown in the following scenarios.

Equipment receives S1F14 from host before sending S1F14

Communications state = NOT COMMUNICATING

COMMENT	HOST		EQUIP.	COMMENT
		<--	S1F13	Establish Communications Request
Establish Communications Request	S1F13	-->		
Reply COMMACK = Accept	S1F14	-->		S1F14 received from host and communications established

Communications state = COMMUNICATING

		<--	S1F14	Reply COMMACK = Accept

Equipment sends S1F14 to host before receiving S1F14

Communications state = NOT COMMUNICATING

COMMENT	HOST		EQUIP.	COMMENT
		<--	S1F13	Establish Communications Request
Establish Communications Request	S1F13	-->		
		<--	S1F14	Reply COMMACK = Accept Communications is established

Communications state = COMMUNICATING

Reply COMMACK = Accept	S1F14	-->		S1F14 received from host

If the S1F13 / F14 transaction is still open when the transition to COMMUNICATING occurred, subsequent failure to receive a reply from the host is considered as a communication fault by the equipment.

The scenarios of establish communications capability of ESEC2007 can be illustrated in the following tables.

Host attempts to establish communications

Communications state is enabled.

COMMENT	HOST		EQUIP.	COMMENT
Establish Communications Request	S1F13	-->		
		<--	S1F14	Reply COMMACK = Accept Communications state = COMMUNICATING

Equipment attempts to establish communications and host acknowledge

Communications state = NOT COMMUNICATING

COMMENT	HOST		EQUIP.	COMMENT
				[LOOP] [LOOP]—SEND
		<--	S1F13	Establish Communications Request
Establish Communications Request Acknowledge	S1F14	-->		
				[IF] S1F14 is received without timeouts [THEN] exit loop –SEND [ELSE] delay for interval in Establish communications Timeout [END IF] [END LOOP] –SEND
				[IF] COMMACK = Accept [THEN] communications state = COMMUNICATING exit loop –SEND [ELSE] Reset timer for delay, and Delay for interval specified in Establish communications Timeout [END IF] [END LOOP]

4.4.2.3 Control

The control state model is used to describe the implementation of this capability. The control capability is initiated by the equipment through pressing on the function keys only. Host can not initiated the control state. Based on this state model, the system activates in either the ON-LINE or OFF-LINE state. Upon entry to OFF-LINE state, the system can activate either EQUIPMENT OFF-LINE or ATTEMPT ON-LINE that are the substate of OFF-LINE. Entry to the ON-LINE state, the equipment can switch between REMOTE and LOCAL substate by reading a function key in the front panel of the equipment. This is to determine the appropriate state.

In the control state model, only the operator affects the control state. The operator retains ultimate authority to set the equipment OFF-LINE by direct pressing on an OFF-LINE key. In the same meaning, the operator also can cause the equipment to attempt to go ON-LINE.

Since the communication is successful established, the operator can request ON-LINE by pressing ON-LINE key. The equipment will send an S1F1 to the host. The host may reply ON-LINE with an S1F2 or deny ON-LINE by sending and S1F0. If the host reply an S1F2 to the equipment, the control state will entry into ON-LINE with LOCAL state and the main window changes to be below.

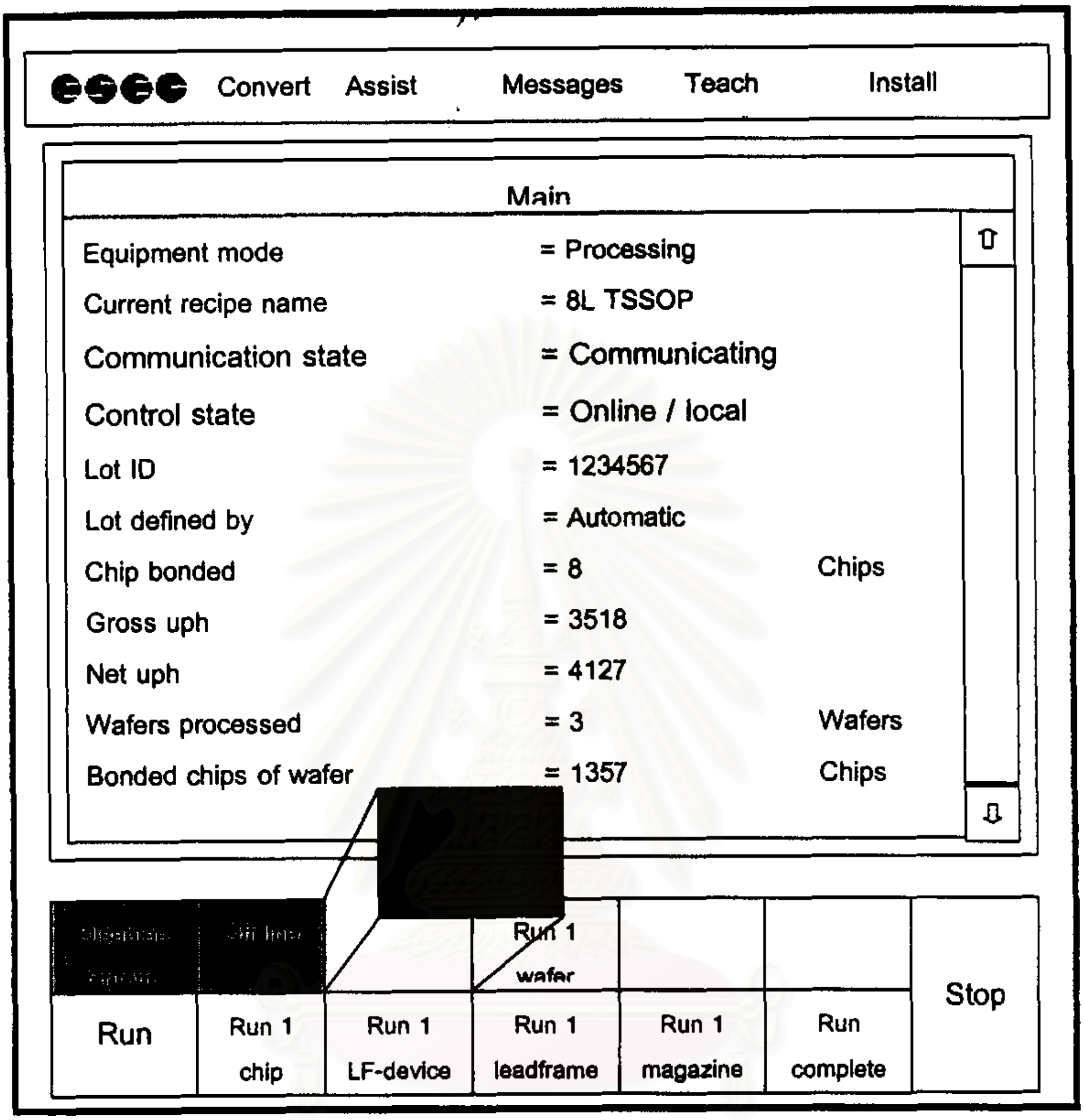

Figure 4.20: Main window and function keys after entry to ON-LINE state.

The scenarios of each case that relate to the control state are shown in the below tables.

Host accepts ON-LINE

Operator actuates ON-LINE switch when OFF-LINE state is active.

COMMENT	HOST		EQUIP.	COMMENT
		<--	S1F1	Equipment requests ON-LINE
Host grants ON-LINE	S1F2	-->		
		<--	S6F11	"Control state LOCAL/REMOTE" event.
Acknowledge	S6F12	-->		

Host denies ON-LINE

Operator actuates ON-LINE switch when OFF-LINE state is active.

COMMENT	HOST		EQUIP.	COMMENT
		<--	S1F1	Equipment requests ON-LINE
Host denies ON-LINE	S1F0	-->		

Operator sets OFF-LINE

Operator actuates OFF-LINE switch when ON-LINE state is active.

COMMENT	HOST		EQUIP.	COMMENT
		<--	S6F11	"Equipment OFF-LINE" event.
Acknowledge	S6F12	-->		

Operator sets LOCAL

Operator actuates LOCAL switch when equipment REMOTE state is active.

COMMENT	HOST		EQUIP.	COMMENT
		<--	S6F11	"Control state LOCAL" event.
Acknowledge	S6F12	-->		

Operator sets REMOTE

Operator actuates REMOTE switch when equipment LOCAL state is active.

COMMENT	HOST		EQUIP.	COMMENT
		<--	S6F11	"Control state REMOTE" event.
Acknowledge	S6F12	-->		

4.4.2.4 Data collection

As previous mention that the events are narrowed down and represented in the term "Collection event" which is significant to the host. Data collection capability allows the host to monitor equipment activities via event reporting. Also, the host can query the selected status or other variable data through this capability.

The data collection is classified into 3 activities based on the purpose and data type.

1) Event data collection

The event data collection provides automatic notification of equipment activities to the host. It is useful in monitoring the equipment status or activities. Knowledge of the occurrence of events relate to the equipment state models (Communications, Control, Processing state model) allows the host to track the equipment state. Thus, this capability helps the host understand how an equipment behave and react to the host.

The ESEC die attach machine model 2007 provides a set of collection events as defined in the **APPENDIX C.** Specific collection events are required by the individual capabilities and state models. The examples of collection events include:

- Selected processing and material handling activities,
- Operator action detected by the equipment,
- A state transition.
- Alarms and exception conditions.

Both the equipment and host can initiate the collection event capability. In case of event occurs on the equipment, the collection event will be sent by the equipment. In the other way, host also can request the event report from the equipment. The scenarios of event data collection capability are shown in the following tables.

Collection event occurs on the equipment

COMMENT	HOST		EQUIP.	COMMENT
		<--	S6F11	Equipment sends event report.
Host acknowledges event report	S6F12	-->		

Host requests event report

COMMENT	HOST		EQUIP.	COMMENT
Host requests an event report	S6F15	-->		
		<--	S6F16	Equipment sends event report.

2) Status data collection

This capability allows the host to query the equipment for selected information. The host queries the equipment's status by specifying the desired SVID's. Therefore, the equipment sends the value of these selected status variables to the host. It is not only the value of selected status variables but also the description of status variables such as name and units can be requested.

The S1F3 is used to request the value of status variables while S1F11 is used for description. The status variables that available on the ESEC die attach machine model 2007 are illustrated in the **APPENDIX D**. The scenarios of status data collection are shown in the following tables.

Request equipment status report

COMMENT	HOST		EQUIP.	COMMENT
Host requests report of selected status variable values.	S1F3	-->		
		<--	S1F4	Equipment responds with the requested status variable data.

Request equipment status variable namelist

COMMENT	HOST		EQUIP.	COMMENT
Host requests equipment to identify selected status variables.	S1F11	-->		
		<--	S1F12	Equipment responds with the requested status variable descriptions.

4.4.2.5 Alarm management and exception reporting

The ESEC die attach machine model 2007 uses exception reporting to notify the host of alarm conditions and abnormal situations that occur on the equipment. In order to provide more extensive and flexible data reporting four collection events are defined for each class of possible alarm condition on the equipment. The exception reports are sent by the equipment using the Event Report / Acknowledge (S6F11 / F12) transaction. The following are the example and meaning of exception events

1) **Material warning**

   A piece of equipment is going to run out of material, but operation can continue for a while.

2) **Material stop error**

   A piece of equipment detects the end of material. This situation will cause to terminate the processing.

3) **Equipment stop error**

An abnormal situation was detected by equipment. It requires operator intervention. This situation causes to terminate the processing.

4) Software system error

A fatal error was detected in the software. No recovery is possible so that the equipment has to be shut down.

5) Hardware system error

A fatal error was detected on the hardware. No recovery is possible so that the equipment has to be shut down.

The scenario of this capability is similar to the data collection with linking exception text to the reports.

Abnormal situation occurs on the equipment

COMMENT	HOST		EQUIP.	COMMENT
		<--	S6F11	Equipment sends event report with exception text.
Host acknowledges event report	S6F12	-->		

4.4.2.6 Material movement

This capability is to notify the hosts whenever material is ready for loading / unloading from / to any of the ports on the equipment. This capability allows the host to coordinate material transfer between the equipment and a transfer agent and to track material at the equipment. The equipment advises host each time when a magazine is transferred to or from a buffer. Thus, event report occurs with standard S6F11 / F12 transaction. The scenarios of the material movement are shown as following.

Equipment loaded/unloaded material from/to a buffer

COMMENT	HOST		EQUIP.	COMMENT
		<--	S6F11	Send collection event to host.
Host acknowledges event report	S6F12	-->		

4.4.3 Error messages

Error messages provide the host with information of the reason for a message or communication fault detected by the equipment. A communication fault occurs when the equipment does not receive an expected message, or when either a transaction timer or a conversation timer expires.

The equipment informs the host if it can not process a message due to an incorrect device ID, message stream type, message function type, message format or data format. Moreover, the transaction timer expires also causes to the error message sending. The scenarios of the error message can be illustrated as following.

Message fault due to unrecognised DEVICE ID

COMMENT	HOST		EQUIP.	COMMENT
Host sends a message.	SxFy	-->		
				Equipment detects an unrecognised device ID.
		<--	S9F1	Equipment reports to the host that an "**unrecognised device ID**" was detected in the received message.

Message fault due to unrecognised STREAM TYPE

COMMENT	HOST		EQUIP.	COMMENT
Host sends a message.	SxFy	-->		
				Equipment detects an unrecognised stream type.
		<--	S9F3	Equipment reports to the host that an "**unrecognised stream type**" was detected in the received message.

Message fault due to unrecognised FUNCTION TYPE

COMMENT	HOST		EQUIP.	COMMENT
Host sends a message.	SxFy	-->		
				Equipment detects an unrecognised function type.
		<--	S9F5	Equipment reports to the host that an "**unrecognised function type**" was detected in the received message.

Message fault due to ILLEGAL DATA FORMAT

COMMENT	HOST		EQUIP.	COMMENT
Host sends a message.	SxFy	-->		
				Equipment detects an illegal data format.
		<--	S9F7	Equipment reports to the host that an "**illegal data format**" was detected in the received message.

Message fault due to TRANSACTION TIMER TIMEOUT

COMMENT	HOST		EQUIP.	COMMENT
				Equipment does not receive an expected reply message from the host and a transaction timer timeout occurs.
		<--	S9F9	Equipment reports to the host that a transaction timer timeout occurred.

Message fault due to DATA TOO LONG

COMMENT	HOST		EQUIP.	COMMENT
Host sends a message.	SxFy	-->		
				Equipment detects that the message from the host contains more data than it can handle.
		<--	S9F11	Equipment reports to the host that "data too long" was detected in the received message.

4.5 Die attach machine error detection system development

The error detection system is developed to monitor the processing state and detect error of the ESEC die attach machine model 2007. As previous mention that ESEC2007 is designed for the host communication interface. The SECS-II messages are transferred between the equipment and host through RS-232 serial communication port. A female connector is mounted on the equipment, therefore, male connector is required on the cable from host computer. The numbers of die attach machine per a host depend upon the numbers of serial port on a host computer. However, the equipment ID must be different. The example of system is illustrated in the following figure.

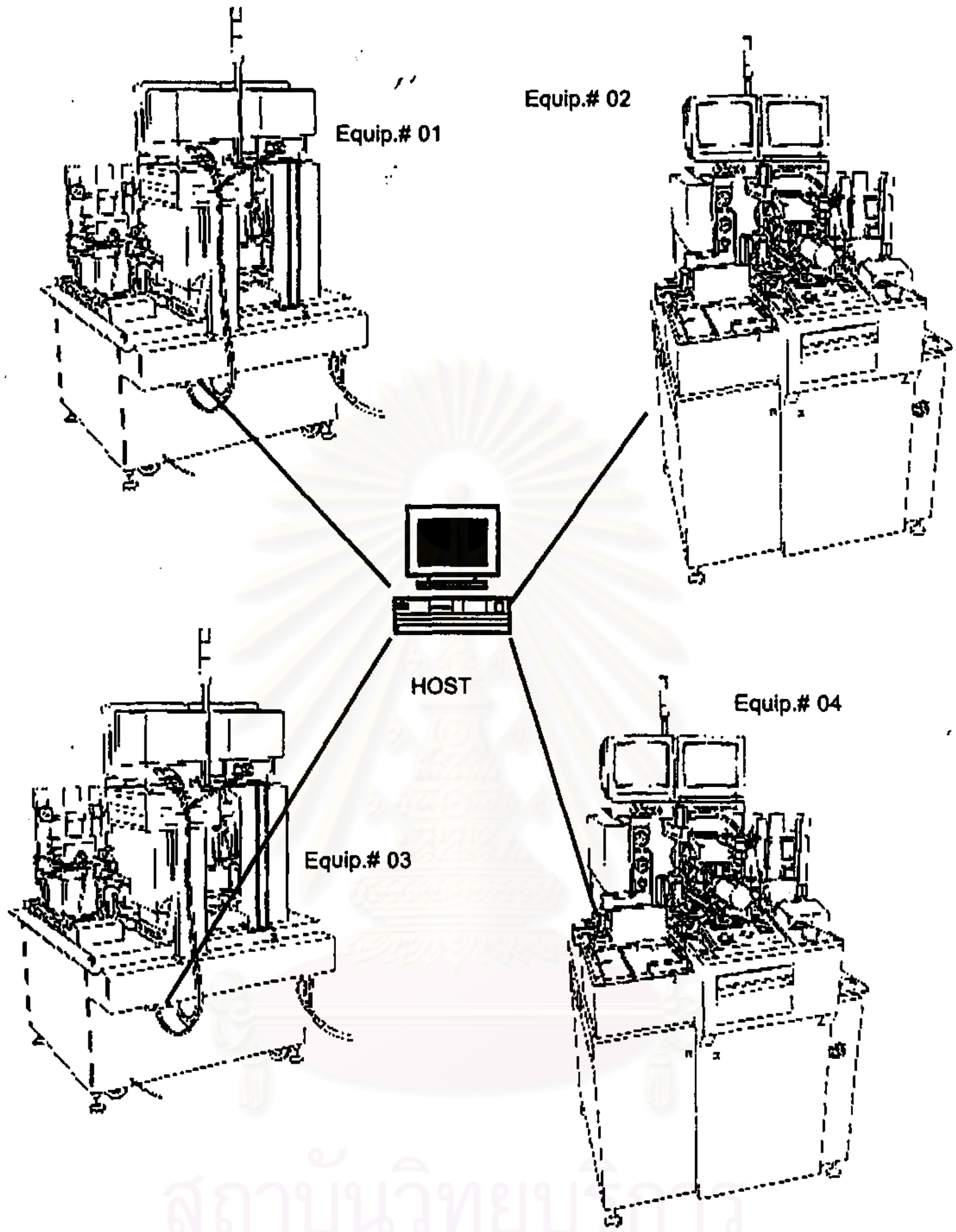

Figure 4.21: Connection diagram of the error detection system.

According to the above figure, the equipments and host are linked together and SECS messages are transferred through serial port. Hence, the host computer can detect machine status and error events.

Details of the die attach machine error detection system on both hardware and software are explained in Chapter 5.