<u>PRIME COMPUTER INTERNATIONAL</u>

REFERENCE NOTES ON THE AMLC

PREPARED BY: C PARTRIDGE

NOVEMBER 1978

CONTENTS

1) INTRODUCTION

This document is designed as an aid to using and understanding the AMLC hardware and software.

The standard documents describe the use of the AMLC related commands, but a description of how the software and hardware works can only be found in internals course notes, which really require attendance on the course.

Many problems occur in normal usage of the AMLC due to a lack of knowledge of how best to use the system. When it comes to making a modification to the software to adapt it for a special requirement, all nature of problems occur.

The information contained in this document is split into a number of sections:

a) A brief description of the AMLC.

b) The user commands and what they do.

c) A more detailed view of the software.

d) Interfacing special devices and coping with known bugs.

e) Differences on the P300.

The information refers to the segmented architecture. The differences in the P300 are described in Section 6.

The details refer to the Rev 15 and Rev 16 releases of PRIMOS.

2) BRIEF DESCRIPTION OF THE AMLC

2.1 The Hardware:

The AMLC (Asynchronous Multiline Controller) interfaces full duplex/half duplex data lines to a PRIME computer.
There are basically three types of boards:

5002, 5004 half duplex
5052, 5054 full duplex
5152, 5154 full duplex with QAMLC

The last digit refers to the number of lines (2 = 8, 4 = 16).
The half duplex type isn't supported by standard software.

A P300 can handle 2 boards (not QAMLC type). A P350, 400, 500 can handle QAMLC with a 400, 500 expandable to 4 boards.

Information is transferred by Programmed Input Output (PIO), interrupt and DMX transfer. PIO is used for setting states or reading control words. Information transfer is achieved on the standard board by DMC on input and DMT for output. The QAMLC board uses DMC for input and DMQ for output. The speed of a line may be altered by software as can the character format and parity.

2.2 The Software

The components of the software for the AMLC are:

a)	The AMLC driver AMLDIM	(Segment 6)
b)	The AMLC phantom interrupt code	(Segment 4)
c)	The user ring buffers	(Segment 7)
d)	The input tumble tables	(Segment Ø)
e),	The dedicated cells	(Segment Ø)

The software uses two basic mechanisms. The first one, DMX transfer occurs without direct software intervention. The second one, interrupt processing involves a) and b).

The design aim is to reduce the overheads incurred with the 2nd mechanism because this software is of course consuming CP power.

2.3 DMX Transfer

This mechanism uses cycle stealing. This means that the flow of execution is not affected while DMX is going on. However, in the micromachine which is where the microcode comprising each instruction is being executed, there is a temporary break to handle the DMX service. This microcode is known as firm wear.

Incoming characters from the device use Direct Memory Control. This method uses a pair of pointers in memory to indicate a memory area where characters can be placed. Each AMLC board has two such pointer pairs and memory areas (known as tumble tables). At Cold Start, the AMLC board (the controller) is loaded with these pointer pairs, and triggered. For a system with 4 boards there are consequently 8 tumble tables. Each tumble table is 48 words long. Characters arriving from a device are routed to the tumble table. The 2 byte (1 word) entry consists of a line number and the character, or a bit pattern in the line number byte to indicate a condition ie: break. This process continues until the tumble table is full. At this point, the controller signals this fact (interrupts) and switches input to the other tumble table. This toggling action continues automatically. It is the responsibility of the software to remove these characters before the toggle action overwrites the table.

Outgoing characters can use one of two mechanisms:

a) DMT (Direct Memory Transfer)

b) DMQ (Direct Memory Queue)

DMT is the most common mechanism. In memory, a cell is maintained for each line. The controller is given the address of the cell block. Each cell is scanned at the rate for the line pertaining to the cell, for presence of a character. If a character is present, it is moved to the output device and the cell cleared by the controller. It is the responsibility of the software to fill the cells at a sufficient rate to satisfy the line speed to which the cell relates.

The second mechanism, DMQ is available on the 51 series boards. With this technique, the dedicated cell is replaced by a queue. It is the responsibility of the software to top up the queue before the AMLC has extracted all the characters at the line speed.

2.4 Interrupt Processing

Transfers to and from memory occur without software interruption. It is the responsibility of the software to remove the characters from the tumble tables at a fast enough rate and place characters in the dedicated cells or queues to satisfy the line speeds. The software is invoked by means of interruption from the controller. Each line on the controller has a flag bit called the Character Time Interrupt flag (CTI). If this flag is enabled then an periodic interrupt is generated by the AMLC at the rate for the line. The worst situation could be every line going at 9600 baud with the CTI flag on. In this case it is unlikely that the CPU would do anything apart from running AMLDIM, trying to service this interrupt rate. This state of affairs is avoided in a balanced system by using the CTI flag in an ordered manner. For input the CTI flag is set on a particular line at a low rate. This nominated line, called the input clock line, (one for the whole system) is set to interrupt 10 times per second

At this rate, software examines the tumble tables and removes the characters. This is fine while the input rate is low (human type speed). A second mechanism exists to handle the case where characters are coming in more rapidly ie: a fast device sending in characters. When a tumble table is full, the AMLC recognises this and generates an interrupt known as an End of Range (EOR) interrupt. This causes the software to clear the tumble table, hopefully before the other tumble table fills, (which, of course, happens normally). These two mechanisms cope with the two extremes. The first one, typing a few characters at one terminal, ensures that the characters are interpreted by PRIMOS and not just left in the tumble table until an EOR is eventually generated. The second one, flooding the AMLC with characters, prevents data loss except in the limiting case where the input rate is greater than the ability of the software to handle it.

For output the CTI flag is set on a particular line at a faster rate than input. This line is called the output clock line, (one for the whole system). For the DMQ case: A single clock line controls output and input. In the DMT case the software examines the dedicated cells of all the lines and fills up any that are zero if characters are available. In the DMQ case, the software tops up the queues if possible. This system is fine if the lines are operating at the output clock line speed (or lower) in the case of DMT. If it is desired to run the line at a high speed, then two techniques are available. The first one is to make the output clock line run at the high speed. The disadvantage of this is that the amount of CP power required to service this rate increases. At 9600 baud the CPU can spend a large percentage of time (>50%) checking the dedicated cells, if this technique is adopted. The second technique is to switch on the CTI flag for the particular line. However when no more characters are to be transmitted, then the flag must be switched off (otherwise the overheads approach the first method).

Normally the second method is adopted. The first one is usually only chosen by accident. With DMQ high speed lines are handled by increasing the size of the queue so that the topping up of the queue 10 times a second can cope with the higher rate. In practise it is difficult to drive a line at the maximum rate of 9600 baud due to machine loading.

2.5 Software Implementation

The previous section described the software mechanisms that are operating system independent. In other words, the interrupt processing is not dependent on the type of operating system. If the system has an AMLC board, then the software must perform the required servicing. This section describes the software conventions adopted by PRIMOS to interface the AMLC to the rest of the system.

The first consideration is the eventual destination of incoming characters and the store where outgoing characters reside. Each configured line (terminal users and assigned lines) has an input and an output buffer. These buffers are circular (ring) and default to 192 characters on input and 384 characters on output. Characters arrive at the input buffer from a device at the rate the device is transmitting. When the buffer is full, echo back is disabled. User space programs remove characters from the buffer using normal input read routines. Characters arrive at the output buffer from user space programs. When the buffer is full, the user is suspended. Associated with each line is a data word called the LWORD. This is used by the software to determine which buffer is being used for the line and various characteristics set for the line.

Note echo is achieved in the software not in the controller.

At cold start time, a test is made to see how many boards are plugged into the system. The internal tables are adjusted according to the result. The last line is called the group 1 line and determines the rate at which the tumble tables are scanned. The next line back is called the last line of group Ø and determines the rate at which the dedicated cells are scanned for output. In a DMQ system, there is no group 1 and the clock line becomes the last physical line.

3) THE USER COMMANDS

This section describes the commands that affect the AMLC and its associated software. The user has to be the supervisor (system console) except for the ASSIGN and TERM command.

3.1 AMLC

This is the major command affecting the AMLC. It is issued from the system console either "on the fly" or in the C ←PRMO file. The format is:

AMLC (protocol) line number (config) (Lword)

The variants are:

- i) AMLC protocol line number config
- ii) AMLC protocol line number config Lword
- iii) AMLC line number config
- iv) AMLC line number config Lword
- v) AMLC protocol line number

The protocol may be TRAN, TRANHS, TTY, TTYHS, TTYNOP. The HS protocols invoke the CTI bit on output. Consequently these are used if the line is being set to a speed greater than the output clock line. For DMQ systems HS must not be used. The difference between TRAN and TTY concerns the treatment of newline characters, the parity bit and echo.

For TTY protocol carriage return is echoed for line feed, bit 8 is set true and the character is echoed unless specified otherwise in LWORD. TTYNOP disassociates the line from a user space and it is used when:

a) A USRASR space is being set up and can be used to achieve:
b) An assigned line is being set up

In case a) the line being no opped is 2 less than the user number. Case b) is usually specified if transparent protocol is being used. The line number is specified in octal. The config word is a bit pattern used to set up line speeds, stop bits and character length. On receipt of the config word, PRIMOS issues a PIO to the controller to alter its state. The speed bits have 4 fixed speeds, a programmed clock and 3 jumper assignable speeds. The programmed clock is usually set to 9600 baud. The jumpers have to be set on a complete board basis. Normally installations choose the intermediate speeds between 1200 baud and 9600 baud. The LWORD controls treatment of carriage return, echo and XON/XOFF. The right hand byte determines whether the line is associated with a user space. To make a line assignable, this byte must be cleared. The exact specification of the config LWORD bit pattern can be found in the System Administrators Guide.

3.2. ASSIGN/UNASSIGN

This command is used when it is required to assign an AMLC line. It is issued from user space. It uses the same format as AMLC, the ASSIGN/UNASSIGN being placed before AMLC, ie: AS AMLC etc.

Two important points to note are:

a) LWORD can not be altered from user space.

b) Not specifying the protocol will default the line to TRAN.

The implications of a) are that features like XON, if set up this way, have to be done on the LWORD attached to the original AMLC command input at the system console. The implications of b) are that if a feature like XON is required, then TTY or TTYHS must be specified because XON will not work under TRAN. For the UNASSIGN, an abbreviated syntax is allowed, ie: UN AMLC lineno.

3.3 AMLBUF

This command can only be issued at cold start from the CONFIG data file. It is used to change the buffer sizes and the Queue size if DMQ is being used. Note, however, that the latter doesn't work under Rev 15. The parameters are octal words, so for buffer sizes, a conversion to decimal characters has to be made, eg: a parameter of 1000 would give a buffer of 1024 characters. The line number is also octal.

Problems occur if AMLBUF is being used to alter assigned lines. The line number must be the next one beyond the terminal lines for the 1st assigned line and the one above that for the next and so on. This is because the buffer given to an assigned line is taken from a pool residing above the terminal buffers. The order in which the buffers are given is determined by the order in which the lines are assigned. The physical line is not used for these calculations. Imagine a system where NUSR = 4 and NAMLC = 3. The AMLBUF command must use line number 3 for the 1st assigned line, 4 for the 2nd and 5 for the 3rd. The line actually assigned is immaterial.

When using the DMQ parameter, the queue size must be calculated 2**K, $4 \geq K \leq 16$ If the queue size is less than 16, then a machine halt will occur.

3.4 NUSR

This command controls the number of terminal lines configured for this session. NUSR must be placed in the CONFIG data file. NUSR which is octal, represents the number of users including the system user.

3.5 NAMLC

This command controls the number of available AMLC lines. Buffers are locked according to the combination of NUSR and NAMLC.

3.6 TERM

This command alters the characteristics of the AMLC from user space. It makes the LWORD bits available at user space, in particular XON/XOFF and duplex. TERM will clear bits 4 - 8 of LWORD so, if these bits have been used by a modified system, then care must be exercised.

4) INNER DETAILS OF THE AMLC SOFTWARE

This section is intended to give an indepth view of the software. If it is required to hang devices on the AMLC or modify the software for specials then the implications of doing this have to be understood so that unpredictable side effects are not experienced.

4.1 Overview

The most important module handling the AMLC is AMLDIM. This module runs as a complete process and has its own semaphores to control the character flow. AMLDIM is where control goes eventually when an interrupt is received. This module uses a number of other modules:

i) FMLIOB (From Logical Input Output Buffer). This module is responsible for obtaining characters from the ring buffer and passing them to AMLDIM.

ii) TOLIOB (To Logical Input Output Buffer). This module is responsible for placing characters in the ring buffer (either input or output).

iii) BUFCHK. This module examines the ring buffer to see if there is room for a given number of characters.

The code that handles the interrupt is contained in SEG 4. This code causes the interrupt response code (IRC) to be invoked.

4.2 Phantom Interrupt Code (PIC)

When an interrupt is received by the microcode, control passes to a location in segment 4. The current PB register and KEYS are saved by the microcode and the code located in segment 4 is executed.

For the AMLC this code consists of 5 instructions. There are 4 OCP instructions and an INEC AMLSEM. The OCP instructions clear the AMLC's interrupt mask and disable any further interrupts. The INEC is a process exchange instruction that:

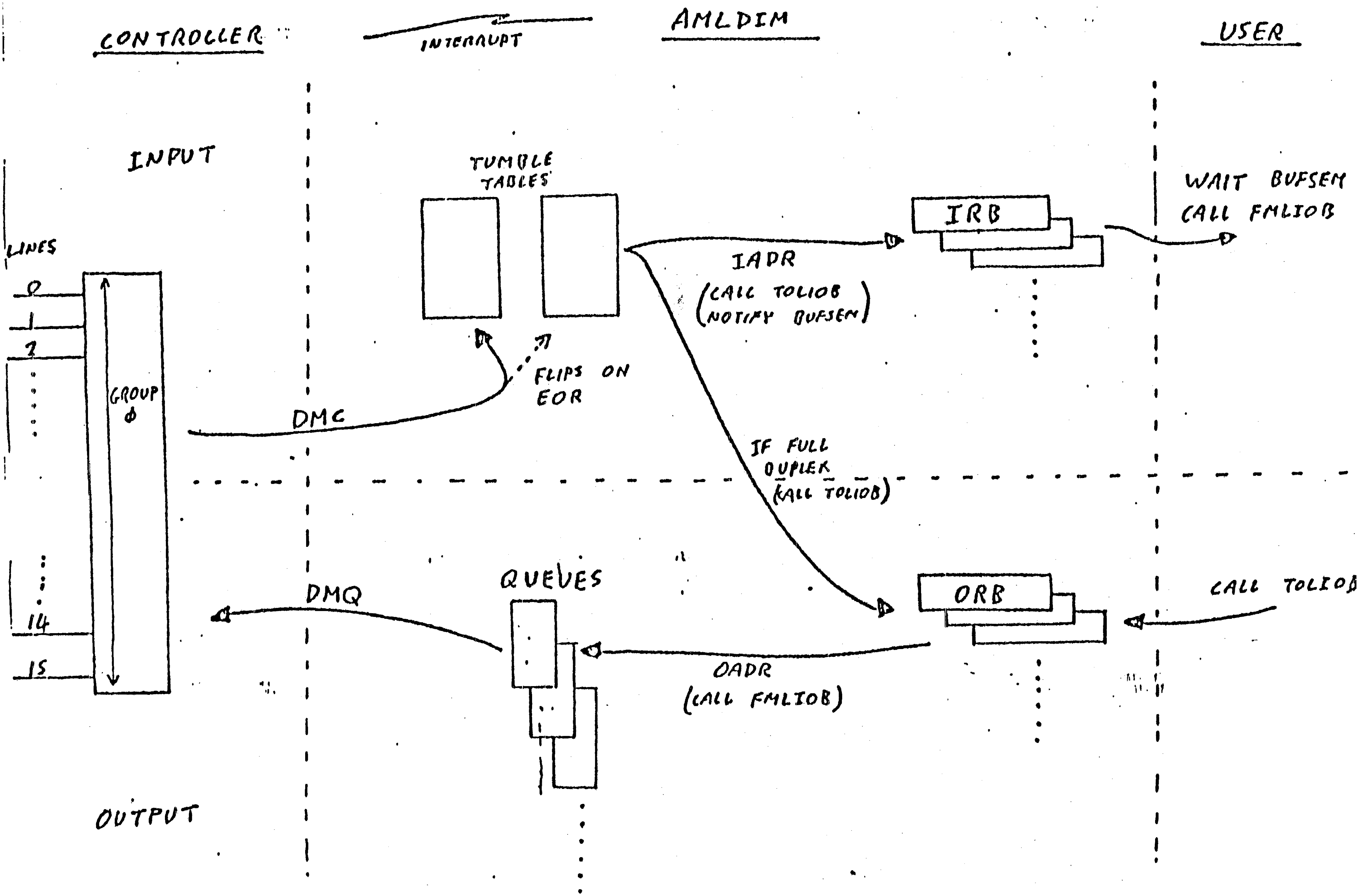
i) Notifies the semaphore AMLSEM and places the PCB on that semaphore on the end of the ready list at correct level.

ii) Issues a CAI operation which frees the backplane of the CPU for further interrupts.

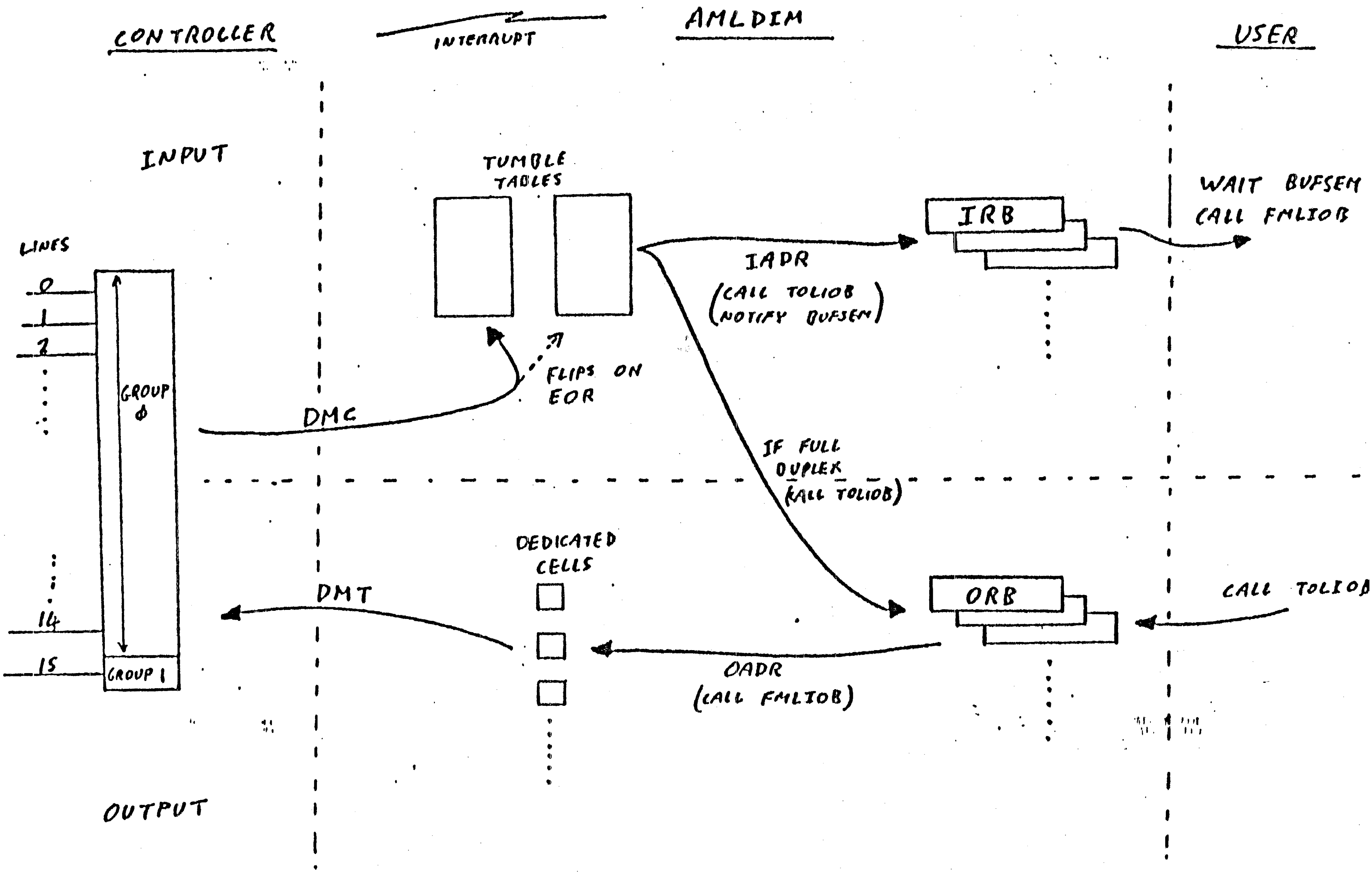
The operation performed in i) means that the AMLDIM process which, in idle state waiting on AMLSEM, gets moved onto the ready list by the dispatcher (a microcode operation). The position it occupies on the ready list is governed by its level, which is 2 for the AMLC. Only the clock and SMLC are higher. The significance of the end positioning means that if other processes were on the same level, then the AMLDIM process would be placed at the end of the chain. However, as AMLDIM is the only process at this level, this is of no significance. The level is set in the PCB at System Startup. The dispatcher then either schedules the new process (AMLDIM) if it is now at the highest level or, else continues with the current process. The latter will only occur if the current process is the clock or the SMLC.

The end result is that the AMLC gets serviced very rapidly. When the AMLDIM process has finished, then the dispatcher schedules the next process in the ready list. This could be the one that was interrupted or a higher one if another interrupt had occurred after the AMLC one.

QAML


CONTROLLER
INTERRUPT
AMLDIM
USER
INPUT
TUMBLE TABLES
WAIT BUFSEM
CALL FMLIOB
IRB
LINES
IADR
(CALL TOLIOB
NOTIFY BUFSEM)
0
1
2
FLIPS ON EOR
GROUP Φ
DMC
IF FULL DUPLEX (CALL TOLIOB)
QUEUES
DMQ
ORB
CALL TOLIOB
14
15
OADR
(CALL FMLIOB)
OUTPUT

AMLC


CONTROLLER
INTERRUPT
AMLDIM
USER
INPUT
TUMBLE TABLES
WAIT BUFSEM
CALL FMLIOB
LINES
0
1
2
IRB
IADR
(CALL TOLIOB
NOTIFY BUFSEM)
GROUP Φ
FLIPS ON EOR
DMC
IF FULL DUPLEX
(CALL TOLIOB)
DEDICATED CELLS
DMT
ORB
CALL TOLIOB
14
15
GROUP 1
OADR
(CALL FMLIOB)
OUTPUT

4.3 Basic Flow Through AMLDIM

Referring to the diagram, the basic flow starts with the dispatcher (microcode) giving control to AMLDIM. After the 1st interrupt, after cold start, the process (AMLDIM) is always on a WAIT instruction. The first task is to identify the controller that interrupted. These tests are performed in Rmode because PIO cannot be performed in Vmode. Any PIO instruction is converted to an EIO which occupies 2 words. Failure to find the interrupting controller causes a HALT. Having identified the interrupting controller, the status word for that controller is input to determine what type of interrupt occurred. Three types of interrupt can occur:

i) End of Range (EOR)
ii) Character Time Interrupt (CTI)
iii) Multiple CTIs

Case i) is indicated by bit 1 being set (the sign bit)
Case ii) is indicated by bit 9. Bits 13-16 indicate the line.
Case iii) is indicated by bits 9 and 10.

Case iii) occurs if a 2nd CTI is generated before the INA instruction is issued to get the status.

If none of these cases is detected then a WAIT on AMLSEM is issued and the dispatcher reschedules another process.

Case i) EOR

Control is transferred to AMLIN. The correct tumble table is located and the table IADR is used to reference the input protocol. IADR has one entry per line which points to a protocol.

The default set up is TTYIN. The AMLC command modifies the table according to the protocol named. The subscript to point into the correct entry of IADR is obtained from the line number held in the tumble table. Control is transferred to the appropriate protocol.

There are two basic input protocols:

- a) TTYIN Teletype input
- b) TRNSIN Transparent input

The purpose of the protocol is to examine the incoming character and make adjustments according to the specification of the protocol. For case a) a test is made to see if its a break character. If not then tests are made to see if XON has been enabled. The character is written to the input ring buffer using TOLIOB and if echo is required then it is also written to the output ring buffer. If the input ring buffer is full, then no attempt is made to write the character away and echo is disabled. Consequently, if the input ring buffer is not cleared, character loss results. For case b) no tests are performed except ignoring break. However, the character will not go to the input ring buffer if it is full.

Both protocols NTFY the semaphore of the line so that a user process waiting on the semaphore will be placed on the ready list.

Even though only one EOR was generated, all the tumble tables are cleared while this scan is being performed. At the end of the loop, the AMLC status is examined back at AMLDIM to see if any other interrupts had occurred (using the same status word containing EOR). If none exist then a WAIT on AMLSEM is issued and the dispatcher gives the CPU to the next user on the ready list.

Case ii) Character Time Interrupt

On detecting a character time interrupt has occurred, a test is made to see which line caused the interrupt. If the line is the input clock line, indicated by its GFLAG being set, then extra functions are perfomred. These are:

i) Testing for loss of carry. The state indicated by a bit in the data set word word for the controller. the DTE (data terminal ready) is dropped for these lines. If carry has been dropped and DISLOG is enabled then an abort flag is set in the process abort word of the PCB. This is done at the half the clock rate (consequently usually 5 times a second). Dropping the data terminal signal for lines that have lost carry.

ii) This occurs every 3 minutes. However, problems occur with this; see section 5).
Every 3 minutes DTR is dropped for all lines that dont have carry. This caters for the case where lines that never had carry, e.g. modem lines, are accidently engaged.

iii) AMLIN is called to clear the tumble tables as for an EOR.

Then AMLOUT is used to examine all the dedicated in the current group (Ø or 1). The mechanism used to do this is to check the output ring buffer to see if any characters exist. If there are characters present then code is entered (depending on the controller type). For the DMT case, the dedicated cell is examined and if it is empty, then the OADR table is used to transfer control to the output protocol for the line. The default output protocol is TTYOUT. Others available are:

a) TRNOUT Transparent

b) TRHOUT Transparent highspeed

c) TTHOUT Teletype highspeed

The main difference exists between the high speed and the normal protocols. The high speed protocols use the character time interrupt bit to over-ride the slower speed of the group clock rate. If there are more than 40 characters in the output ring buffer then the CTI bit is switched on. This of course causes interrupts at the rate for the line. When there are less than 40 characters, the CTI bit is switched off and the dedicated cell is replenished at the clock rate for group zero.

In the DMQ case the queue is examined to see if it can take any more characters. Because DMQ systems do not use high speed protocol, the interrupt is caused by the last line of group zero which occurs at 110 baud.

The routine FMLIOB is used to obtain a character and place it in the dedicated cell for the line or at the bottom of the queue for DMQ.

When all the lines have been serviced, a WAIT on AMLSEM is issued.

Case iii) Multiple Character Time Interrupts

The only difference between ii) and iii) is that the AMLIN loop is executed prior to AMLOUT. This is done becuase there is no guarantee that the multiple interrupt didn't occur on the input clock line. The AMLC status word only contains the line number of the last interrupting line.

5) HANDLING SPECIAL REQUIREMENTS AND KNOWN PROBLEMS

Often it is necessary to interface special devices to the AMLC. It is important to be aware of the consequences of doing this in terms of the effect on the whole system and the effect on the device.

5.1 Known Specials

a) XON/XOFF for input devices

b) Buffered devices for output

c) Page mode devices

d) Cassette Input

e) Adding new protocols

f) Interfacing DMQ boards

a) XON/XOFF

In the standard AMLC software XON/XOFF is supported on output. This means that when the feature is enabled, sending an XOFF to PRIMOS suspends output and sending an XON resumes it. However, some devices used for input, such as cartridge devices, will respond to XON/XOFF. This is designed so that the device can transmit data at high speed with the software stopping the device when its buffers are full. The modification to PRIMOS is fairly simple and involves:

i) Testing when the tumble tables are being cleared to ensure there is enough room in the input ring buffer to hold the data.

ii) If the buffer hasn't sufficient room then placing an XOFF in the output ring buffer.

iii) Testing the state of the input ring buffer if an XOFF had been sent to see if transmission can be re-enabled.

iv) If transmission can be re-enabled, then placing an XON in the output ring buffer.

Invoking special features can be achieved by making use of spare LWORD bits. The main consideration is to ensure that extra code does not increase the overhead in AMLDIM CPU usage. Consequently test i) is the only one that needs to be placed in the interrupt loop. Test iii) can be placed in the low interrupt rate loop eg: carrier loss.

b) <u>Buffered Devices for Output</u>

Some output devices, such as plotters and printers, indicate when their internal buffers are full, by setting an interface line (the busy signal). The standard AMLC 5054 can detect this on pin 8 & make the state of the signal available to the software. Interfacing AMLDIM to these devices can be achieved by:

i) Incorporating a special test in AMLOUT
ii) Adding a new protocol

The modification i) is straightforward but once incorporated, gives the device to a specified line and also involves an overhead in AMLDIM, even if the device is not being used. ii) is a much more satisfactory solution as it is line independent. Care must be exercised when adding this modification that all the precautions are observed when performing the I/O required to read the AMLC status.

c) Page Mode Devices

Page mode terminals are those which transmit a whole screen of information in one burst. This causes a large quantity of information to be sent to the tumble tables. If there are a number of page mode terminals connected to the AMLC, then there is the danger that the tumble tables will not be able to handle the input rate. Consequently, loss of information will occur, which necessitates increasing the size of the tumble tables in segment Ø. The main consideration is to ensure that the disk driver still resides at location 1400. It will also be necessary to increase the size of the input ring buffers using the AMLBUF command.

d) Cassette Input

Cassette input devices are similar to page mode devices, in that they transmit burst mode packets. Consequently the size of the input ring buffers will need to be increased and the tumble tables may need to be increased. If the device responds to XON/XOFF, then the considerations in a) need to be borne in mind.

e) Adding new Protocols

Adding new protocols is a fairly straightforward process. The tables in NLKCOM will need to be adjusted to reference the new protocol name (as input with the AMLC command) to the driver name in AMLDIM. The new protocol code will need to be added to AMLDIM using the logic contained in the existing protocols ie: use of TOLIOB and FMLIOB to manipulate the characters. The only other important consideration is to ensure that the generated code doesn't overflow the page boundries set up in MAPGEN.

f) <u>Interfacing DMQ boards</u>

Adding DMQ boards to the standard system causes no difficulty. The problem comes when a special addition has to be incorporated. The DMQ only affects specials that require suspension of output based on certain requirements. The length of the queue must be taken into account because suspension of transfer from the ring buffer to the queue doesn't affect the DMQ going from queue to the AMLC. It is therefore necessary to pack out the queue with null characters which don't get sent to the device.

5.2 Known Problems

Certain known problems exist which can be got round by using certain techniques.

If forced logout on disconnect is configured (in the CONFIG file) direct connect devices may be logged out. The object is to drop DTR (Data Terminal Ready) on lines with no carrier. However this is done by pretending all lines have carrier. Any line that never had carrier (ie: a direct connected line) will be force logged out. The solution for devices that generate DTR is to use cable type 1470. For devices that do not generate DTR strap DTR from the AMLC to carrier. For the system console being operated as a USRASR terminal, the carrier must appear high on the line that corresponds to the buffer being switched. The alternative is to set the LWORD to zero.

If forced logout on disconnect is enabled, then output may not be turned on. This is because the logout message is attempted before the LWORD is changed to allow output (ie: the buffer number inserted). If the output ring buffer is full then the process (user) hanges on a semaphore. Message all now can cause the ring buffer to fill.

Unstable carrier can cause problems such as random disconnects.

Problems can occur with UK Modems because noise on the line may cause the modem to think carrier is permanently high. Carrier high with no one logged in can cause a modem to become permanently engaged by a wrong number.

The maximum size of all ring buffers (in total) must be less than 32K words.

6) P300 DIFFERENCES

The mechanisms used by the AMLC hardware are independent of system as the same controller is used throughout. The main difference between the P300 and P400 concerns the segmented architecture of the latter.

The AMLC driver AMLDIM doesn't differ significantly between the P300 and P400. The technique of tumble tables, dedicated cells and ring buffers applies. DMQ is not available on the P300.

The most important difference concerns the way the code is entered. As there is no process exchange mechanism, the interrupt address is the entry point for AMLDIM. The DMX memory areas exist in the same segment as the driver. The ring buffers exist in a pseudo segment which is addressed through the memory mapping tables.

The parameters of the AMLC software are fixed and changes can only be made at source level. The most common change is the buffer size. This can be achieved by modifying the module TFLIOB. The main consideration is to ensure that the centronics buffer start address is located on a page boundary.

The suspension of users is achieved by a state vector. This means that if a user requires input, he will not get access to the ring buffer until a time slice interval (unlike PRIMOS IV) where he will be waiting on BUFSEM and get put on the ready list by AMLDIM. This of course has consequences when servicing fast devices.

XON/XOFF is not implemented in the standard system, although insertion of the code is fairly straightforward.