Forth computer

In describing memory and i/o interface circuits surrounding the 6809 microprocessor, Brian Woodroffe introduces more features of his FIG Forth computer in this second article.

The system may be used in partial form. Operating-system and language software exist in eprom, so the computer will work without a floppy-disc drive. Many computers use eprom as a bootstrap to load an operating system from disc, making a disc drive mandatory. Although omitting the disc drive reduces cost by almost half, virtual-memory features of Forth are lost, resulting in a significant degradation of performance. Fewer than one third of the memory devices are essential. Parity-error checking may be omitted. When the system is turned on, it only demands 16K of ram and as more is added the memory map is changed on line, Table 1 (see over).

Circuit description

Memory. Eproms containing fixed instructions of the Forth machine and M6809 peripherals pose few problems. These devices occupy the top 16K memory locations because the 6809 reset vector is in this area and decoding is simple using a dual two-to-four-line demultiplexer i.c. (LS139). Dynamic ram occupies the remaining 48K addresses from 0000 to BFFF. Logic i.cs used to glue the main items together are low-power Schottky devices, chosen for their speed and low power consumption. Standard t.t.l. parts could be used, except in the timing chain for the dynamic rams and on the microprocessor memory and address buses; nmos microprocessor parts have very low driving capability and low-power Schottky inputs require less current than standard t.t.l.

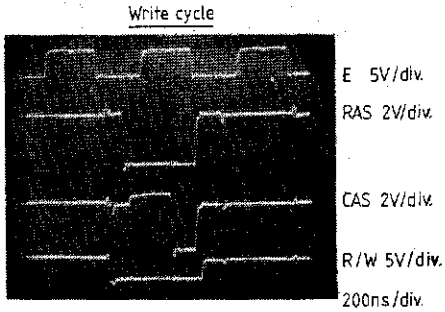
Dynamic rams consist of an X-Y matrix of capacitor storage cells. Access to a bit (storage cell) is gained by first addressing the matrix row. This address is clocked in by the falling edge of the row-address strobe (RAS) and data from all 128 cells in the row are transferred to row buffers. When the column-address strobe (CAS) is true, i.e. low, the column address on the address pins selects one of the row buffers, causing its data to be passed to the output pin. Timing constraints on these actions are fortunately not stringent relative to the time available in a processor cycle.

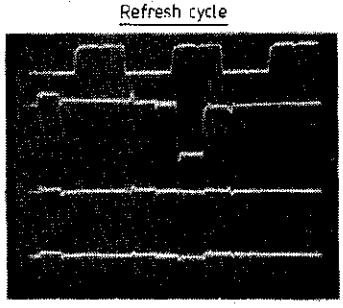
Multiplexing of the 14 address lines onto the seven address pins is done with an I3242 multiplexer. In this design, writing is carried out by the early-write cycle. Within the early-write cycle the write signal is made true before the column-address strobe acts. When CAS becomes true, data on the data input overwrites that of the selected row buffer. Then when the address strobes become false, data from the row buffers are returned to their respective cells, so writing the input data into the X-Y matrix.

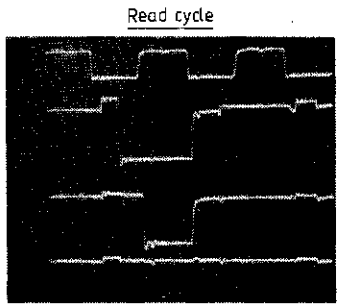
by Brian Woodroffe

Two clocks, E and Q, divide the 6809 processor cycle into four parts. The first quarter of the cycle is used to precharge the rams and as dynamic rams consume most power when the row-address strobe is applied, the selected bank of rams only receives this strobe on the rising edge of clock Q. The address multiplexer is then switched by a delayed Q-clock edge to apply column addresses, leaving sufficeint settling time before the E-clock acts.

During a reading cycle the column-address strobe is made true half way through a cycle (rising edge of E Clock) so that data may be made available by the RAS-selected rams, through the LS245 buffer, to the M6809 before its set-up time. All of the rams receive CAS but only those receiving RAS pass data to the bus.







During a writing cycle data is not made available by the M6809 until the second half of the cycle so CAS is delayed until the falling edge of the Q signal.

Refresh generator

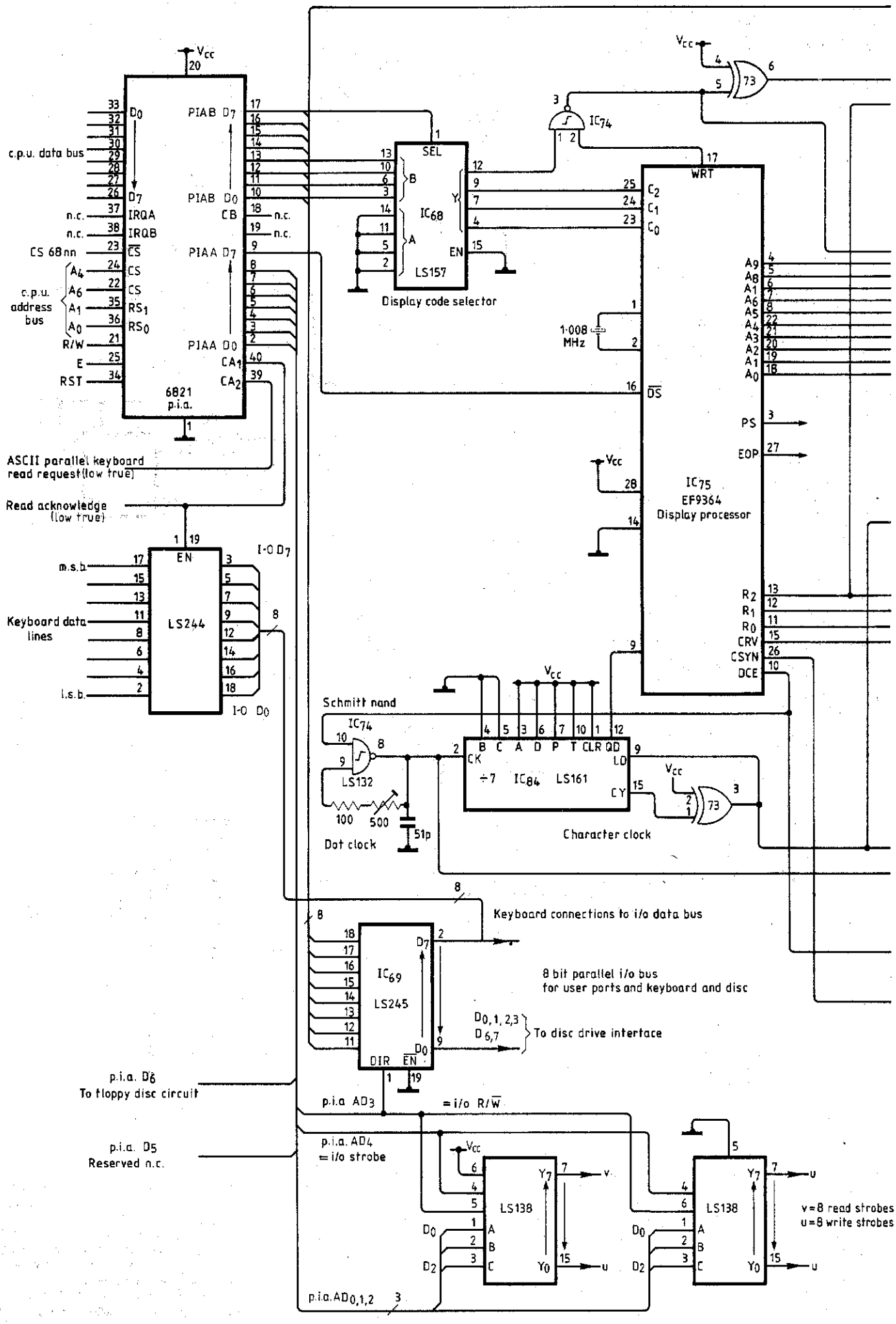
Storage cells in dynamic rams, being capacitors, lose their charge so they must be 'refreshed'. Any memory action refreshes the selected row through data being read into the refresh buffer and returned at the end of the cycle. Unfortunately, program flow will not normally refresh all the ram rows in the allotted time of 2ms and a refresh generator is required.

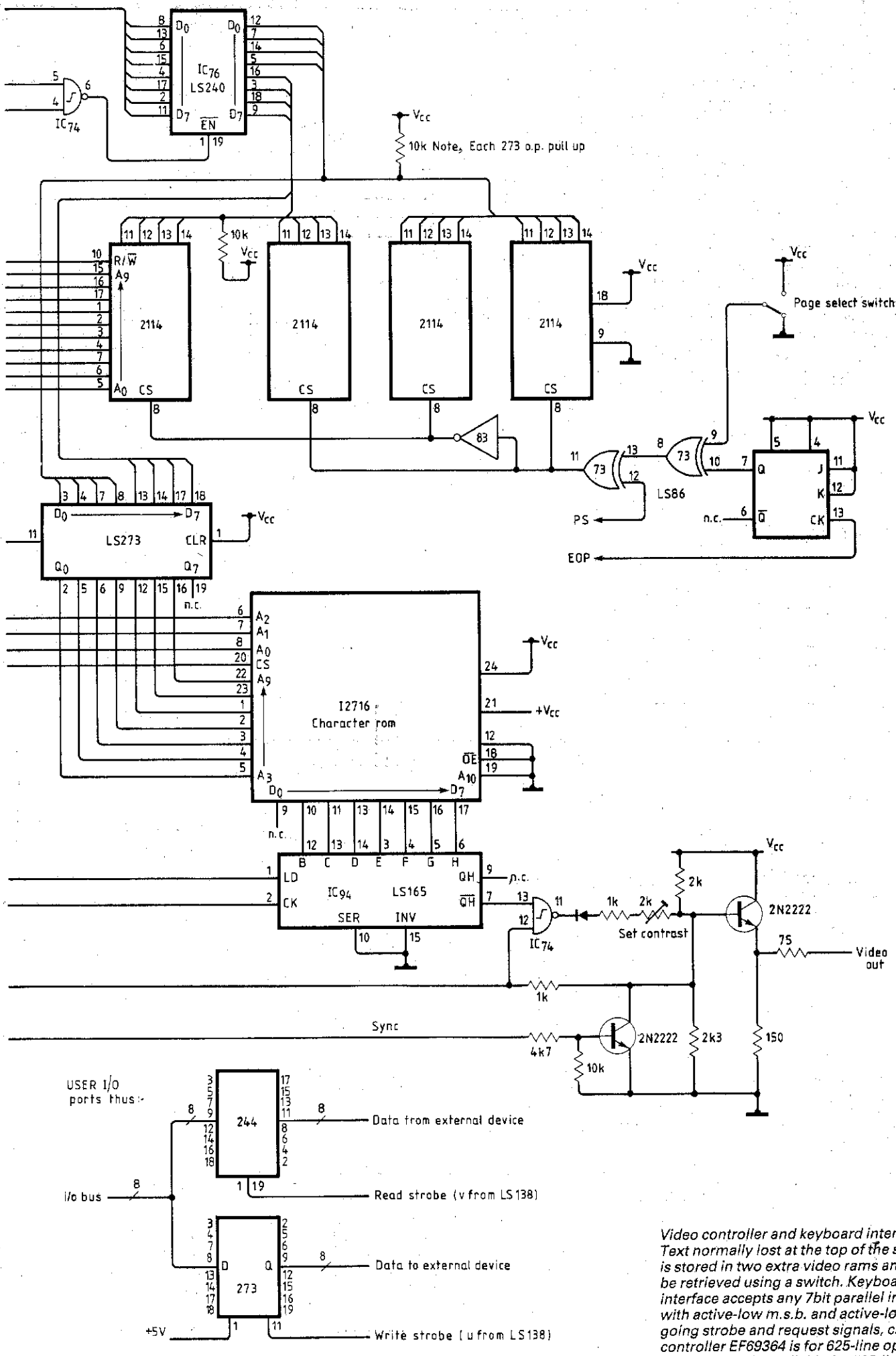
There are three ways of refreshing rams. In burst refresh, normal processor action is suspended and the refresh generator cycles through all 128 rows (for a 16K ram) and returns control to the processor for the remainder of the 2ms. This results in the processor stopping for 128 memory cycles (85μs at the clock speed used). Such a time lapse is unacceptable in this application for the disc drive can require communication with the microprocessor once every 32μs during sector read/write operations.

So, distributed refreshing is required, that is, each successive row is refreshed at 14μs intervals. Distributed refresh generators demand that the processor does not have access to memory while the row is refreshed. The processor may be stopped for this period but a more efficient method is to use a circuit that recognizes when the processor is not using memory and performs what is called a distributed hidden-refresh cycle. This method was chosen.

The refresh generator divides time into 14 cycle quantums using an LS163 counter and generates a refresh-request signal once each period (14μs × 128 cycles = 1.7ms). By monitoring address lines $A_{14,15}$ during the first quarter cycle, the generator knows when the processor does not require access to memory. Having recognized this it generates a refresh-request signal and the I3242 multiplexer places the refresh address on the ram address lines and all row-address signals are set true for a quarter of a processor cycle. During the refresh cycle the column-address strobe is false to inhibit the rams. The address multiplexer advances for the next address and the generator does not demand further refreshes since a flip-flop is set.

It is unlikely that the M6809 will make 14 consecutive memory cycles since all instructions except NOP, SEX and DAA provide non-memory cycles. Should this happen, the refresh flip-flop being reset


c.p.u. data bus
c.p.u. address bus
CS 68nn
R/W
E
RST
n.c.
6821 p.i.a.
PIAB D7
PIAB D0
CB
PIAA D7
PIAA D0
CA1
CA2
IRQA
IRQB
ASCII parallel keyboard read request (low true)
Read acknowledge (low true)
Keyboard data lines
m.s.b.
l.s.b.
LS244
I-O D7
I-O D0
SEL
IC68
LS157
Display code selector
IC74
WRT
1·008 MHz
IC75
EF9364
Display processor
PS
EOP
R2
R1
R0
CRV
CSYN
DCE
Schmitt nand
LS132
100
500
51p
Dot clock
IC84
LS161
÷7
Character clock
Keyboard connections to i/o data bus
IC69
LS245
8 bit parallel i/o bus for user ports and keyboard and disc
To disc drive interface
p.i.a. D6
To floppy disc circuit
p.i.a. D5
Reserved n.c.
p.i.a. AD3
= i/o R/W
p.i.a. AD4
= i/o strobe
p.i.a. AD0,1,2
LS138
v = 8 read strobes
u = 8 write strobes



Video controller and keyboard interface. Text normally lost at the top of the screen is stored in two extra video rams and may be retrieved using a switch. Keyboard interface accepts any 7bit parallel input with active-low m.s.b. and active-low-going strobe and request signals, c.r.t. controller EF69364 is for 625-line operation but a version is available for 525-line tv.

Table 1. Example of how the memory map may be changed when more than 16K of ram is used.

```
FORTH HEX
SMAX DUP @ 4000 + SWAP !          ( allow more data stack)
S0 DUP @ 4000 + SWAP !            ( move data stack)
SP! SMAX DUP @ 4000 – SWAP !      ( reset data stack)
R0 DUP @ 4000 + SWAP !            ( move return stack)
TIB DUP @ 4000 + SWAP !           ( move terminal input buffer)
' FIRST @ DUP @ 4000 + SWAP !     ( move Forth virtual memory buffers)
' LIMIT @ DUP @ 4000 + SWAP !     ( IE 'FIRST' and 'LIMIT')
FIRST DUP PREV ! USE !            ( point virt. memory pointers to virt. memory)
DPMAX DUP @ 4000 + SWAP !         ( move limit of dictionary up)
DECIMAL                           ( return to decimal arithmetic)
```

and the counter carry being set (refresh quantum finished), processor action is suspended by a dummy direct-memory-access cycle which guarantees a non-memory-access cycle.

Parity checking

Capacitance used to store data in dynamic rams is so small that naturally occurring charged particles (alpha particles) have a charge great enough to corrupt data should they hit a cell. Improved coatings on dynamic-ram dies have reduced this effect to give an error rate below 0.1%/1000h for 16K dynamic memories[5]. It is impractical to include error correction in small 8bit memories but parity checking to halt the processor when an error occurs is not.

An odd-parity bit, generated by an LS280 parity checker when a byte is written into memory, is stored with the other eight bits. During the write-cycle the parity-ram data output is in its high-impedance state and the floating EO input is high. The parity device output is clocked into the ram input and correct parity is looked for when memory is read. On reading, the data output drives the parity checker and the error signal is passed to the error latch with the row-address strobe signals. If an error exists, the RAS line concerned is latched, a led indicates which memory bank contains the error, and the processor halts.

Memory speed and drive

Input characteristics of dynamic ram are quite different from those of t.t.l. Ram inputs are capacitive, which especially affects signals common to many inputs like RAS, CAS and WE, and they require little direct current. When driven directly from low-power Schottky t.t.l. these inputs can cause considerable overshoot that can result in exceeding device specifications and longer access times through the time taken for the voltages to level out.

To reduce ringing, some form af matching is required. Series matching is most appropriate since it does not increase static loading. The ideal driver would produce a slightly under-damped response but because t.t.l. drive characteristics are asymmetric a compromise had to be made in the resistance value. Control signals are driven from LS37 clock drivers to ensure adequate drive toward the 5V rail. Resistance values are not critical for this relatively slow memory and the original even worked faultlessly with no damping resistors and standard LS00 drive.

On analysing the timing requirement of the ram/M6809 interface I noticed that the most readily available 200ns rams leave a lot of spare time – so much so that these devices could theoretically be run with a 666ns cycle time instead of the standard 1μs. This was, of course, tried. Not only was it tried with the faster M6809A processor but also with the standard device. In both cases functioning was faultless. This is not to say that all 1MHz parts will run at higher speeds but certainly 200ns access time rams will work at 1.5MHz. So for the cost of a new crystal the through-put of the system was improved by 50%.

Peripherals

To ensure that 1MHz peripheral devices such as the 6821 peripheral-interface adapter and the 6850 communication-interface adapter operate correctly, the memory-ready signal (MRDY) is used. Whenever peripherals are addressed MRDY is held false by an LS122 monostable multivibrator which extends the memory-access time. An M6850 communication device forms the RS232 interface and the clock frequency for it is crystal derived. Currently the 1.5MHz c.p.u. clock only allows 1800bit/s and an external baud generator is an attractive proposition. Both –5 and +12V supplies are used for the RS232 interface. Current from the –5V supply is so low that the RS232 driver has an active current limiter; the +12V drive is resistive.

Many of you will not have an RS232 terminal and will wish to use a separate keyboard and domestic tv. The keyboard interface will accept any 7bit parallel input signal with active-low most-significant-bit and active-low-going strobe and request signals. Two spare hand-shake lines on the p.i.a. and an output port could form a Centronics-type printer port.

An EF69364A video i.c. provides timing signals necessary for a 625-line tv; a 96364B device will provide signals timed for 525-line tv. Control code for the video i.c. is supplied through an LS157 quad two-to-one-line multiplexer and for normal display characters (p.i.a. B D_7=0) a fixed control code is set. When control characters (hexadecimal 0 to F) are used the p.i.a. supplies the relevant code through the multiplexer (p.i.a. B D_7=1) to the EF69364. As the c.r.t. gun scans the screen, the EF69364 selects the character to be displayed from the display ram and latches it into an LS273.

The video i.c. was designed for use with ram that has separate data input and output lines (2101 ram) so the circuit was modified to allow 2114 rams with common i/o to be used. Character-code from LS273 and row information from 69364 is supplied as an address to a character rom (a specially programmed 2716 eprom). Each character position is allocated a 7-wide-by-12-high character block.

Referring to last month's article, the signal name at pin 6 of IC_{41} is active low and should read $\overline{R}$, as should the signal name at the junction of IC_{47} pin 2 and IC_{45} pin 3. On page 57, pins 13, 12 and 5 of the LS175 should be labelled Y_0, Y_1 and Y_2 respectively.

A set of three programmed roms is available from Brian Woodroffe at 632 Queensferry Road, Edinburgh for £23.50 inclusive. Technomatic (see advertisers' index) will supply all i.cs mentioned in this article.

Disc-drive interfacing is described in the next article.

References

5. E. Westfield, Memory system strategies for soft and hard errors, Wescon '79.

continued from page 48

indebted to Keith Frewin, who wrote the SOFTBOX software, for providing roms 385 and 386.

References

1. P. G. Barker, Data Transmission Between Micros, *Electronics and Computing Monthly*, 2(5), 1982, 21-25 and 46-49.
2. P. G. Barker, Introducing CP/M, *Electronics and Computing Monthly*, 1982, in press.
3. A. Osborne and J. Kane, An Introduction to Microcomputers: Volume 2 – *Some Real Microprocessors*, Osborne & Associates Inc, California, 1978, Chapter 10, pp29-49.
4. Commodore Business Machines Ltd., CBM PET 3032N Professional Computer User's Manual, Publ. No. 320856-3, June 1979.
5. Commodore Business Machines Ltd., CBM PET Series 8000 User's Guide, Publ. No. 320894, 1981.
6. MOS Technology Inc., MCS6500 Microcomputer Family Hardware Manual, Publ. No. 6500-10A, 2nd Edition, January 1976.
7. E. Fisher and C. W. Jensen, PET and the IEEE-488 Bus [GPIB], Osborne/McGraw-Hill, California, 1980.
8. National Semiconductor Corporation, *SC/MP Technical Description*, Publ. No. 4200079B, September 1976.
9. I. Williamson and R. Dale, Understanding Microprocessors with the Mk 14, The Macmillan Press, Bristol, 1980.
10. Small Systems Engineering Ltd., 2-4 Canfield Place, London NW6 3BT, UK, SOFTBOX User Manual, Revision 3, 1981.
11. J. Kane and A. Osborne, An Introduction to Microcomputers: Volume 3 – Some Real Support Devices, Osborne & Associates Inc., California, 1978.
12. P. G. Barker, Computers in Analytical Chemistry, Pergamon Press, Oxford, 1982, in press.
13. MOS Technology Inc., KIM-1 Microcomputer Module User Manual, Publ. No. 6500-15B, 2nd Edition, August 1976.