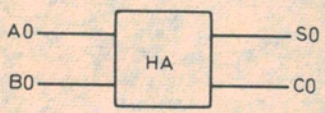
DIGITAL ELECTRONICS BY EXPERIMENT PART 9

Ian Sinclair adds it all up.

Arithmetic Units

SO FAR, THE WORK which we have carried out on the blob-board has covered gating, flip-flops, counter and display stages and the use of a register. Within the limitations of 8 IC's, we cannot, of course, hope to cover every possible principle of digital electronics, and the IC's which were selected for the board were designed to reflect the applications of digital electronics most often seen in published circuits.

The two important topics of arithmetic and memory have not been specifically mentioned, partly because small projects seldom need arithmetic or memory (and large projects can make use of the more flexible facilities of a microprocessor, particularly if this incorporates a memory) and partly because the building blocks of arithmetic units (gates) and some types of memory (flip-flop) have been covered.



A0	B0	S0	C0
0	0	0	0
0	1	1	0
1	0	1	0
1	1	0	1

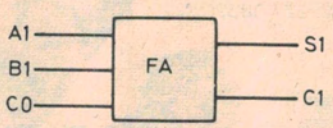
Fig. 1. Half-adder symbol and truth table.

Nevertheless, in this last part we shall look at some of the circuitry we have not covered previously, and also at some systems which can be tried out in the board. In addition, it is useful to note that the board can now act as a very useful intermediate unit for experimental work on more advanced systems, since it can provide up to six clock oscillators, four flip-flops, four NAND gates, one register, and a complete counter circuit and display for one set of BCD digits.

Adding

Binary addition can be *serial* or *parallel*, of which parallel addition is more common. The *half adder* has the truth table in fig. 1 and is used for the least significant digits of two numbers. Its output will be the sum (the digit which will appear in the final figure) and the carry which will be added to the next significant figure. The *full adder* circuit is used for all the next stages of the adder unit and has three inputs and two outputs; its truth table is shown in fig. 2. The inputs to the full adder are the two digits A_1, B_1, and the carry C_o from the previous half-adder stage. The outputs once again are the sum and another carry C_1 which is taken to the next stage. The total number of adding stages which will be needed must equal at least the total number of binary digits in the sum of the numbers.

Half-adders and full adders can be made up using gates (fig. 3) but once the principles have been checked it is easier to use IC's made for the job. The 7482 is a two bit full adder, whose internal circuitry, with truth tables, is shown in fig. 4. From the diagram, we can see that the inputs are C_o from the



A1	B1	C0	S1	C1
0	0	0	0	0
1	0	0	1	0
0	1	0	1	0
1	1	0	0	1
0	0	1	1	0
1	0	1	0	1
0	1	1	0	1
1	1	1	1	1

Fig. 2. Full adder symbol and truth table.

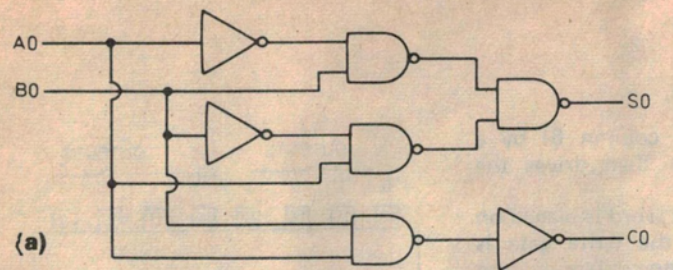
previous half-adder (which would be either an integrated full adder with no carry input, or made up from gates) and the second significant digits A_1 and B_1. The sum of this stage is obtained at the terminal marked S1, and the carry is internally connected into the second stage of the adder, whose inputs are B_2 and A_2 with outputs sum S_2 and carry C_2. The next step up is the 7483, which is a four-bit adder and any requirement greater than this is dealt with by arithmetic units of much greater complexity.

In general, if more than a simple addition is needed, it is more economic to use LSI arithmetic units.

Memories

Memory units which are used in digital work come in several varieties. One class of memory is the volatile memory, based on flip-flops, which is cleared wherever power is switched off; this type could be used in pocket calculators. Non-volatile memories are the types using pre-set registers (such as read-only memories or ROMs) or which use magnetic tapes or cores or other types of storage which are not erased when power is switched off. A simple type of volatile memory is a SISO shift register with its output connected back to its input so that the information is read back in after one complete set of clock pulses; this type of memory can only deliver its contents in the order in which they are stored. If the register has parallel outputs with gates, however, it becomes possible to find which digit (0 to 1) is stored in each flip-flop so that, in the language of computing, random access is possible. This is simple random access memory (RAM).

At this point it is worth pointing out that most memories in general use permit random access. The type of memories which we refer to as RAM are random access memories which can be written as well as read when suitable inputs are applied. They should properly be called random access read/write memories. Read only memories are usually also random access, but the information which is stored has been put there either by the manufacturer (in the design stage) or by the user (as with PROM) when the memory is first used.



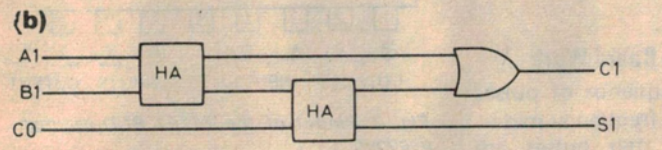
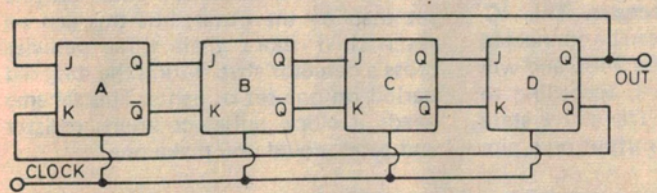


Fig. 3. Above: Adders (a) Half-adder circuit, using NAND-gates and inverters. (b) Full adder, using half adders and OR-gate.

Fig. 4. Right: (a) Schematic of 7482 two-bit full adder. Note again the advantages of medium scale integration. (b) Truth table.

Fig. 5. Below: SISO shift register connected as a memory – the information must be read out in serial form.



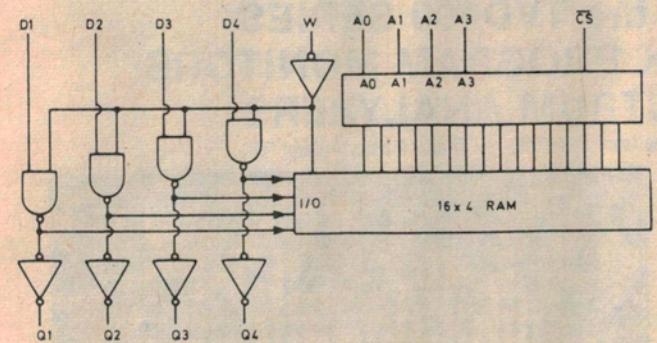
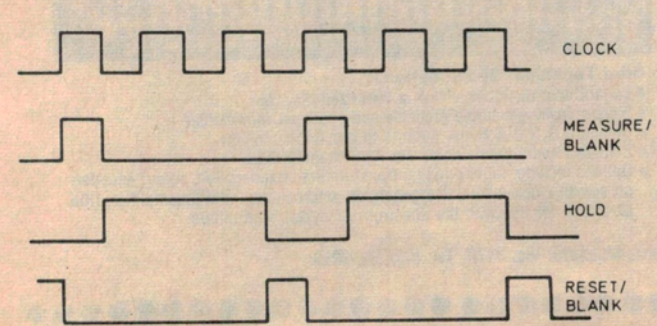
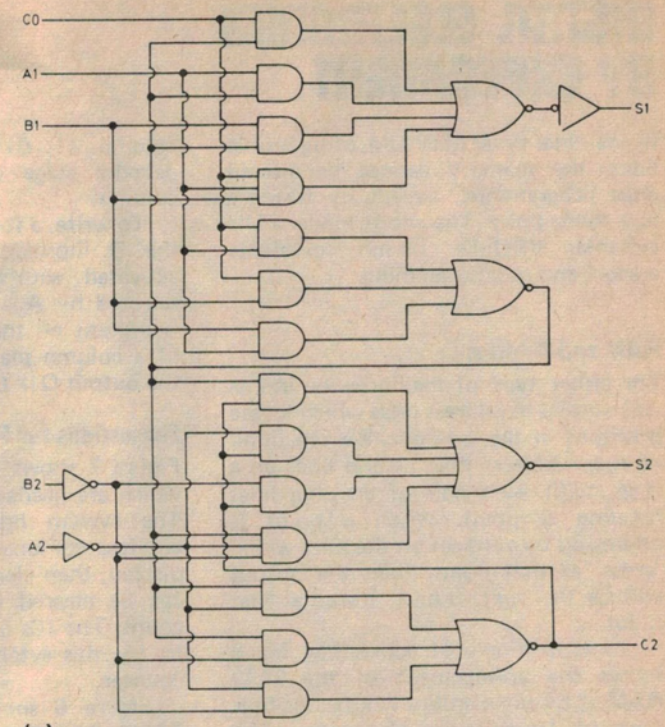


Fig. 6. Above. 7489 RAM schematic, showing addressing system for 16 4-bit words.





TRUTH TABLE

(b)

INPUTS				OUTPUTS					
				C0 = 0			C0 = 1		
A1	B1	A2	B2	S1	S2	C2	S1	S2	C2
0	0	0	0	0	0	0	1	0	0
1	0	0	0	1	0	0	0	1	0
0	1	0	0	1	0	0	0	1	0
1	1	0	0	0	1	0	1	1	0
0	0	1	0	0	1	0	1	1	0
1	0	1	0	0	1	0	1	1	0
0	1	1	0	1	1	0	0	0	1
1	1	1	0	0	0	1	1	0	1
0	0	0	1	0	1	0	1	1	0
1	0	0	1	1	1	0	0	0	1
0	1	0	1	1	1	0	0	0	1
1	1	0	1	0	0	1	1	0	1
0	0	1	1	0	0	1	1	0	1
1	0	1	1	1	0	1	0	1	1
0	1	1	1	1	0	1	0	1	1
1	1	1	1	0	1	1	1	1	1

Fig. 7. Below left: Pulses in a frequency meter. During the measure/blank cycle, the input frequency being measured is gated to the counter, but the display is blanked out. During the hold cycle, the display is on, showing the count, but the input frequency is gated out, so that the reading is steady. On the reset/blank cycle, the counter is reset and the display is blanked. If the repetition rate is more than 50 Hz or so, there is no flicker.

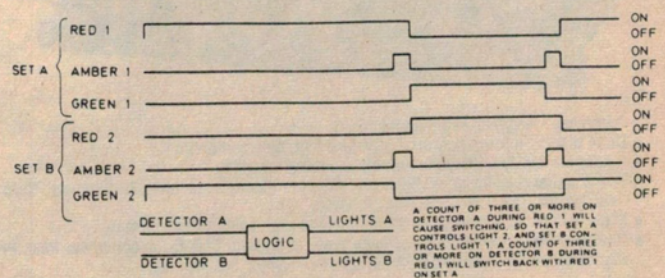


Fig. 9. Above: Priority traffic lights problem. This scheme gives priority (long term period) to the longer line of traffic, as measured by the pulses from the detector pads.

DIGITAL ELECTRONICS BY EXPERIMENT

In the older types of PROM, using fusible links, the memory cannot be altered once programmed, except by fusing a few more links. The more modern UV erasable PROM's permit complete erasure and re-programming.

RAM and Address

For either type of memory, the inputs will consist of address lines which locate positions in the memory. We can think of these address lines as grid lines on a map, with each pair of crossing lines locating a point. When a point is addressed by voltages on the lines which 'cross' at that point, then the output will be the digit, 0 or 1, stored at that point.

As an example of addressing, fig. 6 shows the arrangement of the 7489 RAM, a 64 bit memory which uses four rows of 16 columns of storage. The rows are addressed by the inputs D_1, D_2, D_3, D_4, so that a four bit word can be read into each of sixteen columns. The columns are addressed by another four-bit word which is decoded (1011 = column 11; 0110 = column 6) by a decoder stage which then drives the column.

To write, a four-bit word is placed on the D inputs, and the write gate is activated, with the appropriate column selected by A_0–A_3. To read, no signal is present on the D lines, and selection of a column places a four-bit word on the output Q1–Q4.

Suggestions for Future Board Work

Figure 7 shows the sequence of pulses which are needed by a frequency meter. The system here is that pulses are counted for one unit, count is held on display, then cleared so that the system can be cleared for another (updating) count. The ICs on the board enable you to try this sytem out for one digit of counter.

Figure 8 shows the pinout of the 74141 BCD-decimal decoder. This IC, not used on our board, can be connected to the BCD output of the 7490 and will give outputs on ten pins, according to the state of the count. The active state is represented by a *zero* output on a pin,

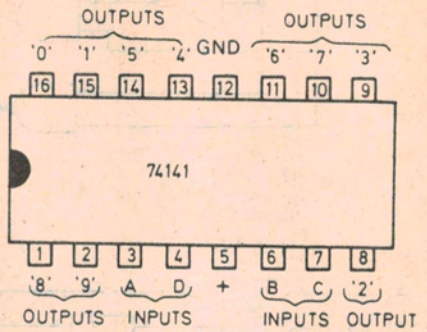


Fig. 8. Pinout of the 74141 BCD-decimal decoder.

so that a zero output on the '7' pin (pin 10) represents a count of 7, and so on. Using this, could you design a ten-note jingle player?

Finally, fig. 9 shows the operation of priority traffic lights. These lights operate with a longer red phase on one set than on the other, but this can be reversed if more than three vehicles cross a detector strip during the long red period on one set of lights. This scheme needs a clock pulse, counters, register and gates, could you make one?