Synchronisation and arbitration circuits in digital systems

D.J. Kinniment, M.Sc., Ph.D., C.Eng., M.I.E.E., and J.V. Woods, M.Sc., Ph.D.

Indexing terms: Multi-access systems, Time-sharing systems

Abstract

Synchronisation of two independently clocked processor units, or arbitration between two asynchronous units requesting access to a common resource, can cause serious time losses in a computer system. The ways in which these problems arise are considered, and a theoretical basis for calculation of the time losses is presented. The theory is then correlated with measurements on practical devices, and currently available methods for minimising the time loss are evaluated. Conditions necessary for prediction of the performance of synchronisers and arbiters are established and it is shown that design principles exist which allow the construction of systems with known reliability.

1 Introduction

In an asynchronous computer system it is possible for two or more autonomous units to request access to a common resource within a short time of one another. This situation occurs typically when a number of independent processors request access to a shared memory as shown in Fig. 1*a*. If the requests are sufficiently widely spaced in time for each to be recognised and dealt with before the next request occurs, then the resource can be allocated as requests occur. However, in the general case, requests may occur at any time relative to one another, and there may be any number of requests outstanding at a given time. Here the resource must be allocated to each of the requesting units in some order which may depend on a predetermined or a dynamically variable priority. An arbitration network now decides the order in which requests are serviced, and this is shown in Fig. 1*b*, which also indicates the request and acknowledge signalling required between the units.

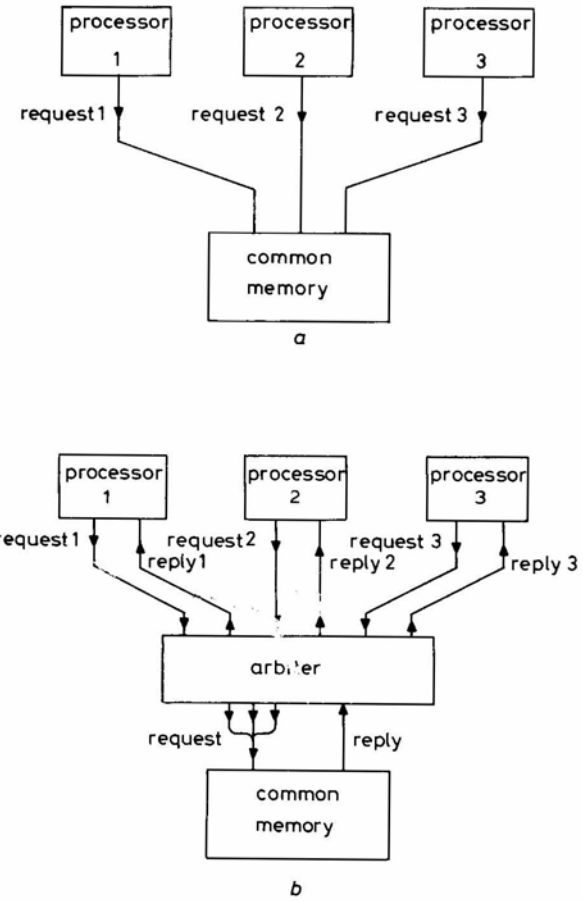

Fig. 1
Access request

a Clashing of requests
b Arbitration and signalling to avoid clashes

To achieve a high level of performance in systems where a memory is shared between processors, it is important to perform the arbitration function as quickly and simply as possible, since a long decision time adds directly to the memory access time, and may also increase the effective memory cycle time. Serious problems can arise in fast asynchronous systems because of the number of decisions which have to be taken, and the time required in some cases to decide whether or not a request exists.[1–3,5–8] In such a system several units may be actively requesting a particular resource, or may be changing state from idle to active at the time the decision is to be made. At that time the unit having priority must be determined on the basis of the request signals outstanding, which may change, and some other external data determining priorities. Any electronic circuit used to select one request out of many will take a finite time to operate, and the request signals must therefore remain steady for the period of time necessary to make the resource allocation.

The design of the arbitration network thus requires the solution of two problems:

(i) the resolution of the time-varying request pattern into a digital vector, constant over the decision period
(ii) the allocation of the resource to a particular request and organisation of the timing.

The solution of the first problem in an asynchronous environment requires special circuit techniques, since it involves the resolution of an infinitely variable quantity (the time of the request) into two discrete states (occuring before or after a particular instant). This problem is closely related to the synchronisation of an external interrupt with the internal clock of a synchronous processor. Here again, the time of an interrupt must be determined as occuring before or after a clock pulse. In either case the time of occurrence of the external event is infinitely variable, and it will always be possible to find a value for the time that causes long response times in any practical decision circuit.

Once a solution to the fundamental decision problem has been proposed within the overall design constraints, the organisation of resource allocation and time can be done by a network of conventional digital circuits designed to suit the particular environment. The design of such arbitration networks for digital systems has received increasing interest[3,4,9,10] with the growth of multiprocessor systems and large fast systems comprising a number of asynchronous units. The purpose of this paper is to present and compare some of the alternative decision circuits available to the system designer, and to examine the design principles involved in the organisation of resource allocation.

2 Decision circuits

2.1 Flip-flop decision times

The function of the decision circuit in an arbitration network is to examine a time varying set of request signals at one particular point in time, and to hold them steady long enough to establish the request with the highest priority. The most commonly used method of achieving this objective is to strobe the request signals into a register, which it is assumed will not change value after the removal of the strobe signal.

Paper 7751E, first received 4th February and in revised form 25th June 1976
Dr. Kinniment and Dr. Woods are with the Department of Computer Science, University of Manchester, Manchester M13 9PL, England

This is shown in Fig. 2 which indicates the situation that may occur if a request status changes just before the termination of the strobe. Here the effective drive pulse to change the corresponding flip-flop within the register may be infinitestimal, and, at the termination of the pulse, the flip-flop output voltage may be at some point between 0 and 1 logic levels.

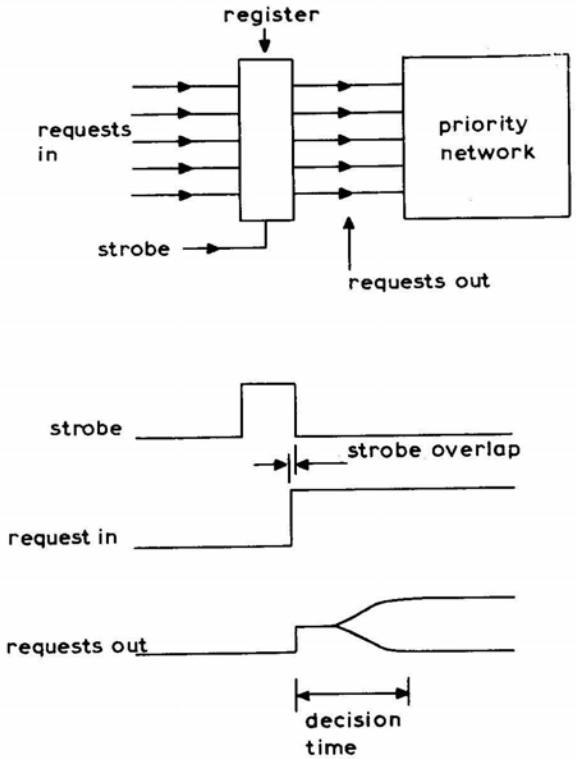

Fig. 2
Decision using flip-flop

Recovery from these points depends on the initial conditions and the characteristics of the flip-flop circuit, but in general the rate of change of the output is proportional to initial displacement from a position of unstable equilibrium midway between the 1 and 0 levels. Recovery from an initial condition exactly on the point of equilibrium can take a relatively long time.

Analysis of the decision time required has been undertaken by several authors,[3,6,9] and the results are summarised and extended here. It is usual to simplify the problem by assuming that the positive feedback of the flip-flop at its position of unstable equilibrium causes a rate of change of output voltage V_0, proportional to the actual output voltage displacement from the equilibrium or zero position. This is true of simple circuits such as a tunnel diode flip-flop which can be represented by a negative resistance in parallel with a capacitance, and approximately true of an e.c.l. flip-flop, a simplified circuit of which is shown in Fig. 3. Here the end of the strobe pulse causes current to be removed from the pair of transistors T_3 and T_4 and to appear in the cross-coupled pair T_1 and T_2. In this situation the base-collector

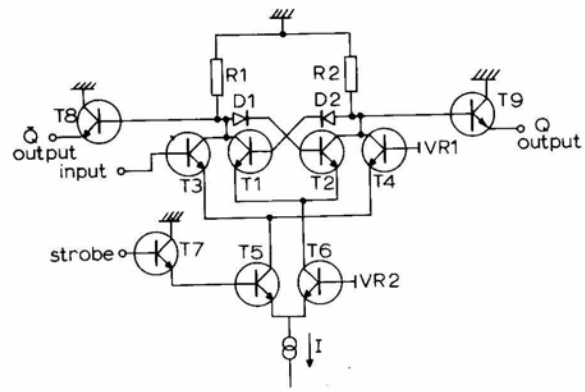

Fig. 3
Simplified e.c.l. latch flip-flop

capacitances of T_1 and T_2 will charge at a rate depending on the voltage difference between the two collectors, and the rate of change of the output voltage difference V_0 will be given by

$$\frac{dV_0}{dt} = \frac{V_0}{\tau} \tag{1}$$

A response characteristic of a positive exponential form will then result in

$$V_0 = Ke^{t/\tau} \tag{2}$$

where the constant K is the initial displacement and τ is a time constant which can be associated with the circuit itself. This model is accurate only in the region where the flip-flop can be represented by a linear system, but as the situation which causes difficulty is one in which the output voltage remains near zero for a long time, the model can give a sufficiently accurate prediction of the response of simple circuits as will be shown later.

In the situation indicated by Fig. 2, the input is represented by logic levels $\pm V_L$, whose overlap in time x with the strobe may have any value. The effect of this on the flip-flop can be evaluated by noting that the feedback loop is broken, but for small values of V_0 the circuit capacitances are now charged at a rate proportional to the input voltage (as the input here is compared with a reference of value 0).

The rate of change of V_0 under these conditions is given by

$$\frac{dV_0}{dt} = \pm\frac{V_L}{\tau} \tag{3}$$

thus the initial condition at the end of a strobe overlap of duration x is given by

$$K = \pm\frac{V_L}{\tau}X \tag{4}$$

From eqns. 2 and 4 it is now possible to predict the time t taken for the flip-flop to respond to a strobe overlap of x seconds to give a total voltage difference of $\pm 2V_L$ at the outputs:

$$t = \tau \log_e \frac{2\tau}{x} \tag{5}$$

To calculate the time which must be allowed for the decision in a given situation, the strobe frequency and the time between successive requests which may change the state of the flip-flop must be known. Assume that the time between strobes is T_s and the time between requests is T_R. The number of times a request occurs x seconds before a strobe termination during a long period of N seconds is given by

$$P = \frac{Nx}{T_s T_R} \tag{6}$$

so that the decision time of the flip-flop will exceed

$$t = \tau \log_e \frac{2\tau N}{T_s T_R} \tag{7}$$

on average only once during N seconds. It should be noted here, that τ is the time taken to move from the zero, or centre point of the output levels to one logic level under the influence of an input and a strobe; it will be approximately half the normal rise time, which is the time taken to move the full logic swing.

2.2 Effect of noise

It can be deduced from eqn. 4, that if each of the range of input request times has an equal probability, then each of the initial condition states will also have an equal probability. Thus the probability P_i of the initial conditions being within the range $\pm V_L$ at some time during N is given by putting $x = 2\tau$ in eqn. 6.

$$P_i = \frac{2N\tau}{T_s T_R} \tag{8}$$

and is a constant with respect to output voltage. The subsequent output voltage trajectories are positive exponentials as shown in Fig. 4.

The probability of the initial state being within the range ΔV between points K_2 and K_3 is given by

$$P_V = \frac{2N\tau}{T_s T_R}\frac{\Delta V}{V_L} \tag{9}$$

and in the absence of disturbance due to noise will be equal to the probability of the output arriving between K_2' and K_3' after a time Δt.

As

$$K'_2 = K_2 e^{\Delta t/\tau}$$

and

$$K'_3 = K_3 e^{\Delta t/\tau}$$

the probability density is thus independent of output voltage, and the addition of a random noise voltage contribution between 0 and Δt will not alter its characteristics as the noise is just as likely to move a trajectory nearer to zero as it is to push one away.

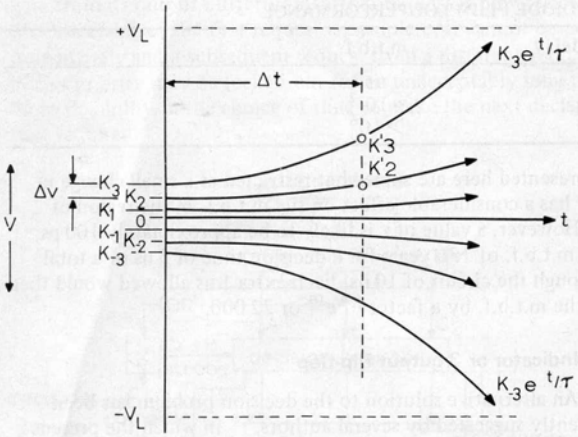

Fig. 4
Output voltage trajectories

The probability of finding a trajectory still within the logic levels after a decision time t will therefore be

$$P = \frac{2N\tau}{T_s T_R} e^{-t/\tau} \tag{10}$$

or alternatively, during a period N seconds the decision time will exceed t on average once if

$$t = \tau \log_e \frac{2\tau N}{T_s T_R} \tag{11}$$

2.3 Experimental

Experimental results illustrating that the probability of a tunnel-diode flip-flop escaping from the metastable region follows the relation

$$P = 1 - e^{-t/\tau}$$

have been published by Couranz and Wann.[9] Results published earlier by one of the authors[6] were obtained on two types of e.c.l. flip-flop by means of the experimental apparatus shown in Fig. 5. This apparatus feeds the clock input of a flip-flop with a strobe signal from an oscillator at a period T_s, and the information input from a separate asynchronous oscillator at a period T_R. After a decision time t the output of the flip-flop is stored and compared with its output a long time later. Selected outputs from an e.c.l. flip-flop still in the metastable region after different decision times are illustrated in Fig. 6. Note here that the discrimination level of FF2 will always be consistently to one side of the equilibrium output level of the flip-flop under test. The apparatus therefore only detects half of the cases where the flip-flop is in the metastable region, and only those output trajectories which finally end on the opposite side of the discrimination level will be shown.

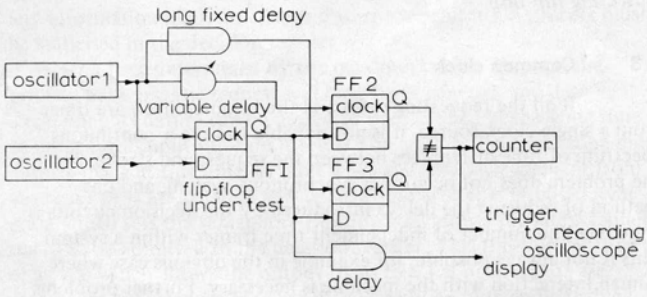

Fig. 5
Experimental set up

Results are presented in Fig. 7 for $T_s = T_R = 200$ ns and two different e.c.l. flip-flops. The propagation delay and rise time of these flop-flops are comparable and the experimental results can be compared to the theoretical predictions by putting

$$\tau_1 = 1{\cdot}1 \text{ ns}$$

$$\tau_2 = 0{\cdot}9 \text{ ns}$$

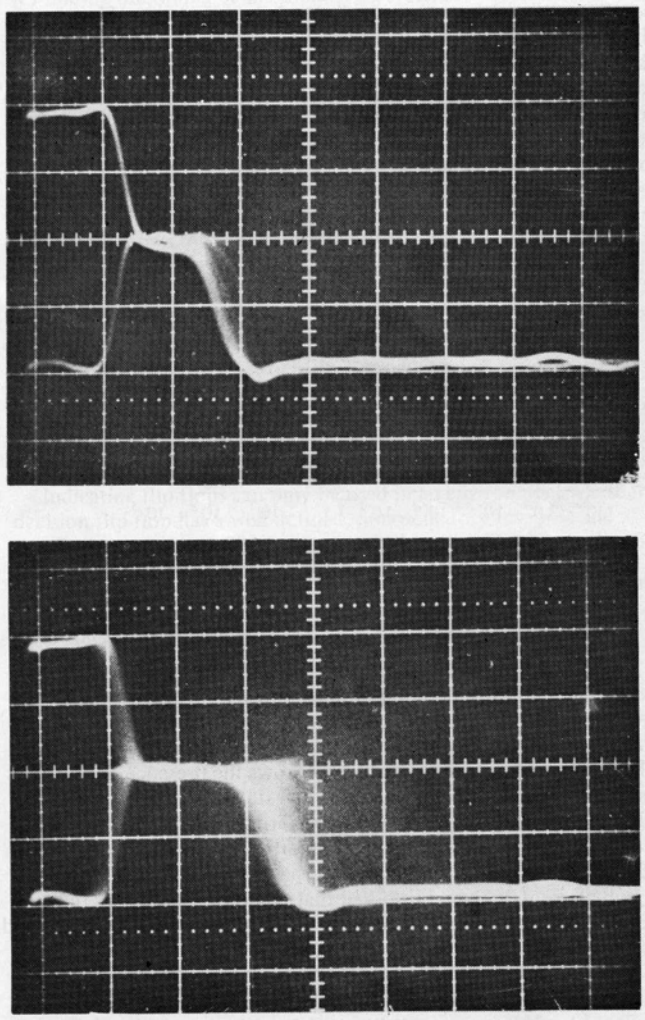

Fig. 6
E.C.L. flip-flop responses

a Delay time at 10 ns
b Delay time at 15 ns

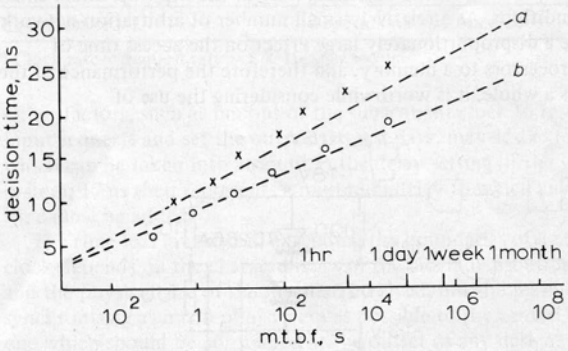

Fig. 7
M.T.B.F. against decision time – e.c.l. flip-flops

a Flip-flop A $\tau = 1{\cdot}1$ ns
b Flip-flop B $\tau = 0{\cdot}9$ ns
x, 0 measured values
---- theoretical characteristic

and m.t.b.f. = $2N$, as nonequivalence in the experiment will only occur once on average in a period of $2N$ seconds. Good correlation is shown between the theoretical and experimental results thus enabling design calculations to be carried out in a practical situation.

2.4 T.T.L. circuits

The development of the formulas in Section 2.1 depends on the assumption that the small-signal response of the circuit under investigation is dominated by a single time constant. This assumption does not hold for a flip-flop constructed from a pair of cross-coupled t.t.l. gates, however. To predict the performance of these gates under conditions where the data and strobe are asynchrnous, it would be necessary to investigate their small-signal characteristics. However, these characteristics are still not enough to predict performance on their own as it is possible for the open-loop gain at high frequencies to be greater than unity, and in some devices even greater than the zero-frequency gain. This leads to output trajectories with a large component of an oscillatory nature superimposed on the positive exponential, a characteristic demonstrated by Chaney and Molnar[7] in 7400 gates. If the oscillation is sufficiently large the operation of the circuit becomes nonlinear, and analysis correspondingly complex.

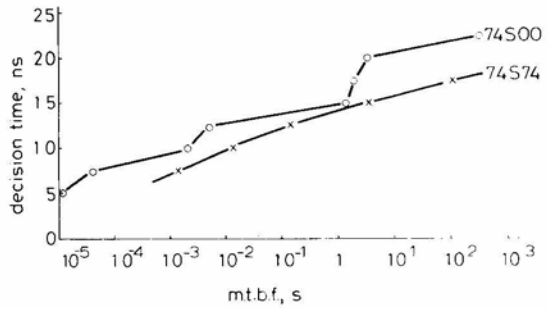

Fig. 8
M.T.B.F. against decision time – t.t.l. flip-flops

The effect of oscillation is shown in Fig. 8 which indicates the performance of a 74S00 cross-coupled gate system and that of a 74S74 flip-flop in which the basic decision element is a simple pair of transistors cross-coupled on the surface of the integrated circuit chip. The cross-coupled gate system clearly shows the presence of an oscillation at a period of 6 ns (twice the gate propagation delay) which significantly deteriorates the performance, whereas the 74S74 circuit performance curve is much smoother. In comparing the time constant indicated by this curve with the propagation delay of the circuit element, it should be noted that the 74S74 comprises the decision flip-flop in series with a second flip-flop which effectively adds to the propagation delay. The performance of the decision flip-flop is thus comparable to that of the 2·2 ns e.c.l. flip-flop.

3 Improvement of the decision time

3.1 Tunnel diode decision circuit

The foregoing results show that, to achieve adequate reliability in a digital system, the decision time allowed for a simple flip-flop is long compared with the normal propagation delay, and is strongly dependent on the switching time of the flip-flop under small-signal conditions. As a relatively small number of arbitration networks may have a disproportionately large effect on the access time of several processors to a memory, and therefore the performance of the system as a whole, it is worthwhile considering the use of

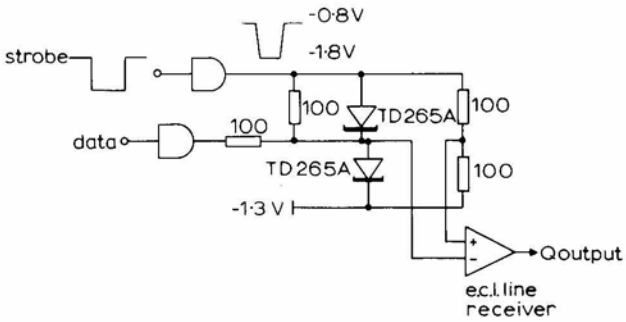

Fig. 9
Tunnel-diode flip-flop

high-performance high-cost units to replace the simple flip-flops. A device which allows fast switching as a flip-flop is the tunnel diode, and units are available with switching times as low as 35 ps.

The circuit of an e.c.l. compatible flip-flop using TD265A tunnel diodes in a Goto pair configuration is shown in Fig. 9. The total decision time is now increased by the e.c.l. input gate for the strobe, and the line receiver used to convert the Goto pair output back to e.c.l. levels. The total delay through these gates is approximately 7 ns, and experiments on the same test apparatus used for the e.c.l. flip-flops give the results indicated in Table 1.

Table 1
TUNNEL DIODE FLIP-FLOP PERFORMANCE

variable delay	m.t.b.f.
ns	s
7·5	$\approx 10^{-2}$
10·0	$> 10^{6}$

Results presented here are somewhat restricted as a small change in the delay has a considerable effect on the m.t.b.f. in the region of interest. However, a value of τ is likely to be approximately 100 ps, giving an m.t.b.f. of 136 years for a decision time of 3 ns or a total delay through the circuit of 10 ns. Each extra 1 ns allowed would then increase the m.t.b.f. by a factor of e^{10} or 22 000.

3.2 Indicator or 3-output flip-flop

An alternative solution to the decision problem has been independently suggested by several authors,[5,8] in which the presence of the flip-flop output trajectory within two voltage limits $\pm Vt$ is detected by comparators, and indicated as a signal on a separate output which holds up further action until the output trajectory has escaped from the metastable region. A circuit of this type is shown in Fig. 10. With this circuit the probability of the trajectory returning to within the measurement band for a significant length of time after it has left depends on the probability of a random noise pulse approaching the magnitude of Vt. Though this is more likely than a change of state of the flip-flop under quiescent conditions, as the width of the measurement band is by definition less than the total logic swing, it remains an extremely unlikely event.

The majority of decisions do not cause entry into the metastable region as shown in eqn. 8, and hence only a few will hold up the internal clock sequence to any extent. This system therefore allows most decisions to be relatively fast but has some disadvantages:

(i) the decision time is now variable and can theoretically approach infinity
(ii) the indication of a metastable state takes some time, typically 7·5 ns in e.c.l. compatible with the 2·2 ns delay flip-flop, and it is necessary always to wait for this time before examining the flip-flop outputs
(iii) it cannot be used if the flip-flop output has any pronounced oscillatory component.

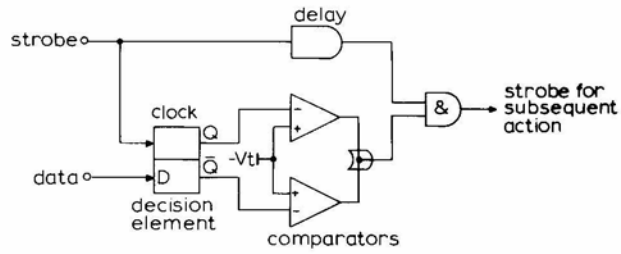

Fig. 10
Indicating flip-flop

3.3 Common clock

If all the requesting units and the resource itself are timed from a single clock source, it is not possible to have a continuous spectrum of time differences between the request and strobe. Thus, the problem does not occur in a synchronous system, and one method of reducing the delays introduced by the decision circuits is to reduce the number of independent time frames within a system. This is not always possible, for example in the obvious case where human interaction with the machine is necessary. Further problems may be caused by the difficulties of accurate clock distribution in a large machine, which can itself cause system delays greater than those introduced by the decision circuits.

4 Arbitration

If a request pattern can be guaranteed stable to an acceptable reliability, it is possible to assign the common resource to one particular request by means of a network of conventional digital circuits and various methods for the design of such networks have been proposed.[4,10]

These methods usually allow only a fixed priority to each request and often assume the arbiter is free to make another allocation as soon as a reply has been received from the resource. The reply, however, may be only an indication that the input data to the resource has been recognised and that the requesting unit is free to remove the data from its output buffer. If a further request is now selected before the processing of the first request is complete, it cannot be processed immediately and a subsequent request from a different source at a higher priority may be locked out for an unacceptably long time. More flexibility in the choice of time to make the next decision is thus required.

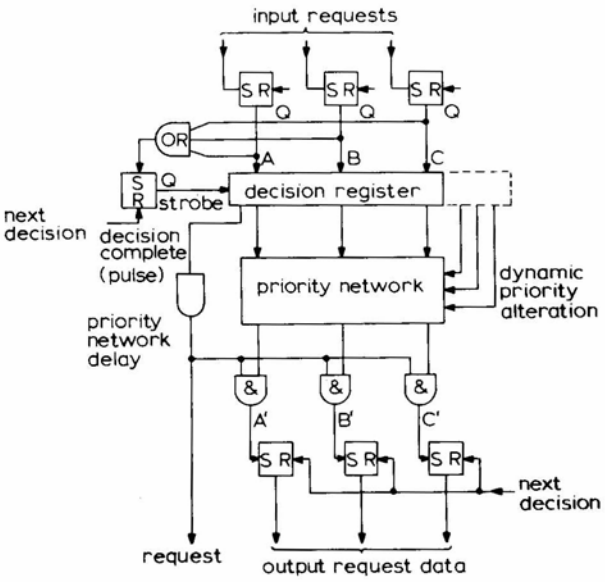

Fig. 11
Arbiter

A block diagram of an arbiter intended to overcome these difficulties is shown in Fig. 11. Here the incidence of one or more requests causes a semaphore flip-flop to be reset, whose function is similar to the P/V semaphore described by Dijkstra[11] in relation to software systems. Resetting this flip-flop causes the strobe to be removed from the decision register which, either after a fixed delay or by means of a combination of indicating outputs, produces a pulse signalling completion of the decision process. At this point the request status is stable until the semaphore is set again later in the control sequence when a further decision is required. The selected request can now be chosen by a simple combinational network, so that only one output of the network is active for any input pattern. Here a fixed priority algorithm could be implemented, or one which is alterable according to the status of the control, for example. If the common resource is a memory with an output register dedicated to read requests, it may be desirable to allow write requests when the register is full, but block further read requests until the data has been transferred to its destination. In this case the request type must be indicated to the priority network as well as the control status, and any information likely to change during the arbitration process must be staticised in the decision register.

After a time determined by the maximum delay through the priority network, the request will have been selected, and data concerning its nature can be passed to the resource at the same time as the request input flip-flop is reset. The reply to the requesting unit may be made at this time, or later when the acknowledgement from the resource is available. A new cycle of the arbiter is initiated by a pulse which sets the semaphore flip-flop, and allows the latest request status to be admitted to the decision register. This flip-flop must be of a type which allows the output to go to a 1 if both inputs are present, as one or more requests may already be outstanding which will have already reset the semaphore thus initiating a new decision process.

This design allows the next decision to be taken at a point determined by the control itself in releasing the semaphore, and the priority algorithm alone determines the order in which decisions are taken if there is a clash.

5 Comparison of decision circuits

Three basic methods have been presented for the improvement or elimination of the long decision times encountered when two or more asynchronous digital systems interact, and these are

(*a*) use of a common time frame for all interacting systems
(*b*) use of a decision element with a fast switching time
(*c*) linking the timing of the interacting systems by means of an indicating or 3-output flip-flop.

Each of these methods is appropriate to certain areas of digital design, for example, a system in which the central clock has a fixed period which cannot be shared by other units such as a fast disc memory with a pre-recorded clock track, cannot use an indicating flip-flop because of the variable nature of the decision time. In general, however, the choice of method depends on the time loss incurred by the use of that method and the system reliability obtainable.

Table 2 shows the time loss involved using different methods in one particular large computer system and illustrates the factors involved in the choice of method. The use of a simple flip-flop for the decision element with an appropriate control delay to give an acceptable reliability results in a time loss depending on the reliability. (Note here that an m.t.b.f. of 10^{12} seconds is much greater than the normally accepted component reliability.) Replacing this flip-flop with a state-of-the-science tunnel-diode decision element reduces the time loss to 10 ns at an m.t.b.f. of 10^6 seconds and the rate of increase of m.t.b.f. with delay time is much improved.

Indicating flip-flops can only be used in an environment where the decision flip-flop has a well defined, nonoscillatory response and a significant time delay may be experienced in an arbiter situation where many requests are competing. Several factors contribute to the time loss as follows:

(i) typical flip-flop delay	2·2 ns
(ii) comparator and gating delay to produce indication of metastable output	5·0 ns
(iii) time cost in combining several indicating outputs for a register	5·0 ns
(iv) clock hold-up gate	5·0 ns
	17·2 ns

Table 2
ARBITER TIME LOSS

	m.t.b.f.* = 10^6 s (11·6 days)	m.t.b.f.* = 10^{12} s (31 700 years)
	ns	ns
Simple flip-flop and delay (normal device delay 2·2 ns)	30	45
Tunnel-diode flip-flop and delay (normal device delay 35 ps)	10	11
Indicating flip-flops (normal device delay 2·2 ns)	minimum of 17 ns	
Common clock	depends on system size	

* Clock rate = request rate = 5×10^6 per second

Other factors, such as fan out of the subsequent clock to reset the input requests and set the output request data, may add extra time which can be taken into account in the delay setting of the simpler system; 17 ns then represents a minimum delay to which any settling time must be added.

The time loss involved in extending the boundaries of a common clock depends on the characteristics of the clock distribution system and the physical size of the machine involved, but the feasibility of synchronising as much of a system as possible to the central clock is one which should be considered at the outset of any design.

6 Conclusions

There is at present a lack of information on the characteristics of flip-flops usable in the decision element situation, and in particular some circuit elements specifically intended for the synchronisation operation are not sufficiently well specified. It has been shown in

this paper that a flip-flop circuit to be used as a synchronisation or arbitration element must have a response which is fast and non-oscillatory. In addition, information on the circuit's small-signal response in the metastable region is required to predict its behaviour. If these conditions are fulfilled, design principles exist that allow the construction of systems of known reliability.

7 Acknowledgments

The work described here arose as a direct result of the MU5 project. The authors would like to thank all members of the University and ICL involved in the project for many helpful discussions, and in particular K.C. Johnson of ICL for suggestions in connection with the indicating flip-flop.

8 References

1 CATT, I.: 'Time loss through gating of asynchronous logic signal pulses', *IEEE Trans.*, 1966, **EC-15**, pp. 108–111
2 LITTLEFIELD, W.M., and CHANEY, T.J.: 'The glitch phenomenon'. Computer System Laboratory Technical Memo. 10, Washington University, St. Louis, Mo., Dec. 1966
3 COURANZ, G.R.: 'An analysis of binary circuits under marginal triggering conditions', Computer System Laboratory Technical Report 15, Washington University, St. Louis, Mo., Nov. 1969
4 PLUMMER, W.W.: 'Asynchronous arbiters', *IEEE Trans.*, 1972, **C-21**, pp. 37–42
5 CHANEY, T.J., ORNSTEIN, S.M., and LITTLEFIELD, W.H.: 'Beware the synchroniser'. Presented at the IEEE Computer Society Conference Comp CON–72, San Francisco, Calif., 1972
6 KINNIMENT, D.J., and EDWARDS, D.B.G.: 'Circuit technology in a large computer system'. Presented at the joint IERE–IEE–BCS conference on computers – systems and technology, London, 1972
7 CHANEY, T.J., and MOLNAR, C.E.: 'Anomalous behaviour of synchroniser and arbiter circuits', *IEEE Trans.*, 1973, **C-22**, pp. 421–422
8 PATIL, S.S.: 'Bounded and unbounded delay synchronisers and arbiters'. MIT Computation Structures Group Memo. 103, June 1974
9 COURANZ, G.R., and WANN, D.F.: 'Theoretical and experimental behaviour of synchronisers operating in the metastable region', *IEEE Trans.*, 1975, **C-24**, pp. 604–616
10 CORSINI, P.: 'Self-synchronising asynchronous arbiter', *Digital Processes*, 1975, **1–1**, pp. 67–73
11 DIJKSTRA, E.W.: 'Co-operating sequential processes' *in* GENUYS, F. (Ed.): 'Programming languages' (Academic Press, New York, 1968), pp. 43–112