

Maximum-rate pipeline systems

by L. W. COTTEN

*National Security Agency
Fort Meade, Maryland*

INTRODUCTION

There is widespread opinion that we are fast approaching the physical limit in speeds for computers. The grounds for such conclusions are traceable to signal propagation delays in interconnections, delays encountered in traversing several levels of combinatorial logic, and to systems organizations. Clearly, these areas must be addressed if we are to realize phenomenal improvements in computer logic speeds over the next decade. Subnanosecond logic circuits will be available; however, design innovations are needed to exploit this performance at the systems level.

Trends in the organization of current high-performance systems (Parallel bits \times clock rate $\geq 10^9$ bits/sec.) have tended toward three types of parallelism. These are arrays of processor elements, processing of operand bits in parallel, and pipelining¹⁻⁶ or compaction of operations in the time domain. In the processing of blocked operands or data arrays^{7,8} all three are potentially complementary rather than competitive. Since most large problems permit iterative processing of blocks of operands, pipelining becomes attractive because the rate of doing work can be increased considerably without a corresponding linear increase in hardware cost. The pipeline queue is the result of organizing logic into an assembly-line process where output rate is independent of the latency or total delay of the pipeline. At any given time a number of operands are in various stages of processing, thus the additional hardware usually takes the form of temporary storage and process control.

The purpose of this paper is to explore the problem of maximum-rate pipeline operation from a viewpoint that differs markedly from current practice. The results will be termed maximum-rate pipelines to distinguish from conventional pipeline systems. In truth, conventional pipelines as presently designed are a special case of the maximum-rate pipeline which constitutes the

general case. To clarify, all pipeline systems in existence today are based on a rate which is the reciprocal of the delay through an elemental pipeline section, typically consisting of a parallel register and perhaps three levels of logic. In contrast this paper proposes a rate of operation largely dependent on the delay difference of the various logic paths through the pipeline section. Since this difference delay can be made considerably less than total delay, higher theoretical rates are possible. Indeed, such rates are surprisingly independent of both propagation delays through $N-1$ levels of gating between clocks, and pure signal transmission delays caused by distances between gates. As will be shown, the maximum rates are limited by uncertainty or variability of paths, and basic circuit switching rate. The latter limit has long been recognized.⁹ It is hoped that the following treatment will provide helpful insight and some design directions that will prove useful in overcoming some of the problems^{10,11} in the high-performance environment.

As a guide to readers, the next two sub-topics will discuss conceptual aspects of maximum-rate pipelines, while the remaining sub-topics will consider the complexities of implementation. The latter subjects may be selectively examined depending upon individual interests. While 50 MHz floating point arithmetic units have been designed using the conventional pipeline concept, no maximum-rate design results are available; however, a numerical example to aid understanding is given at the end of the discussion on logic level variability.

A maximum-rate pipeline queue

The validity of maximum-rate pipeline systems, as with pipeline systems in general, is dependent upon the ability of designers to relate data processing problems to an assembly-line job queue. Feedback is sometimes permitted to slightly reduce average rate, but in general

can usually be deferred in time at hardware expense. For example, this problem has been investigated in the case of accumulators.¹² Since pipeline systems designers must demonstrate the ability to resolve out feedback dependencies, and multiple path interactions will be considered later, it would appear that attention could now turn to the case of a unidirectional pipeline. In the interest of brevity and intuitive understanding, a physical analog of an elementary queue offers some utility.

Consider the conventional pipeline, Figure 1, case 1. Distance X is defined to be interstation separation, and does not include the finite station width. The propagation velocity V between stations is assumed to be positive in the direction shown and non varying. S is defined to be service time at each station, and includes the time needed to traverse the station width. Service

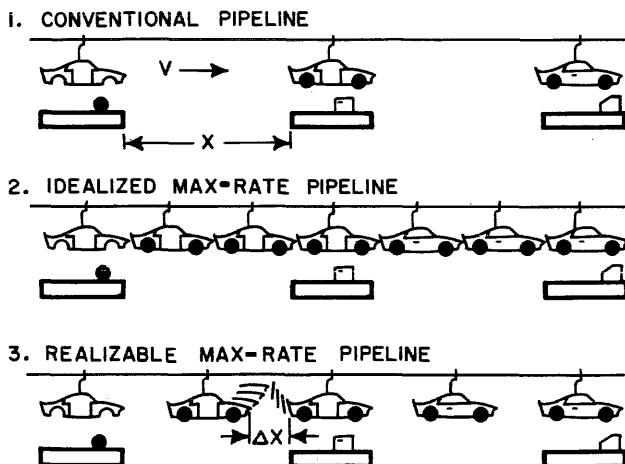


Figure 1—Pipeline queue examples

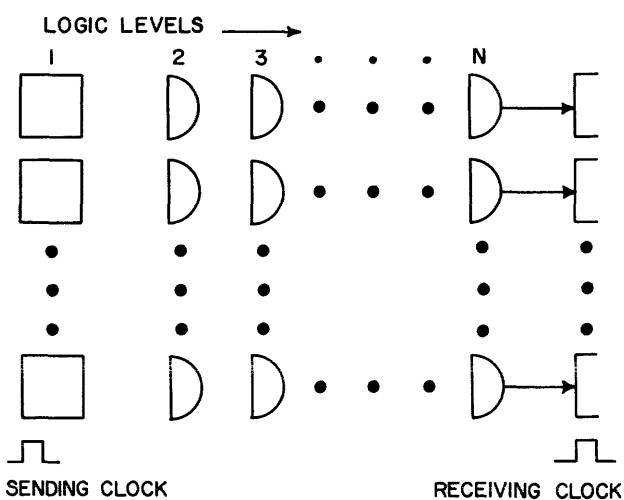


Figure 2—Generalized pipeline section

time S is analogous to data sampling and temporary storage in logic pipelines, and X/V corresponds to propagation delay through logic circuits and interconnections between temporary storage latches, as seen in Figure 2. Under these conditions throughput rate is:

$$R_1 = 1/(S + X/V) \quad \text{units/second} \quad (1)$$

The rate R_1 summarizes the present level of sophistication in pipeline machine design. Borrowing heavily from the notion of a communications channel containing many information symbols or bits in transit, one is enabled to visualize the idealized maximum-rate pipeline, Figure 1, case 2. If S includes time consumed in traversing the finite station width, then ideal rate R_2 is:

$$R_2 = 1/S \quad \text{units/second} \quad (2)$$

Ideal rate R_2 is impossible to realize in practice, because of symbol interference caused by path eccentricities. In theory, the rate approaches $1/S$ as eccentricity is made to approach zero.

In practical circumstances the eccentricity is usually predictable within acceptable bounds, and can be compensated for in Figure 1, case 3, by the addition of a separation ΔX . Under assumptions that the receiving station can undergo minor repositioning or phasing relative to the sending station, the rate becomes:

$$R_3 = 1/(S + \Delta X/V) \quad \text{units/second} \quad (3)$$

If V is a universal constant, the velocity of light for example, its effect could be minimized by reducing the quantity $\Delta X/V$, where ΔX is intersymbol separation as distinguished from X , interstation separation. The time S in (3) is equivalent to the width of one Nyquist interval,^{*13,14} and $1/S$ corresponds to the rate of completeness property which represents the maximum input sequence rate for which any finite state machine can be built.^{15,16}

A maximum-rate pipeline section

The main result of this paper is to apply the max-rate queue concept to a generalized max-rate pipeline section which may be joined with other max-rate or compatible-rate sections to construct max-rate systems. The design of the generalized max-rate section is related to two characteristics of the continuous data stream,

* The Nyquist interval is the minimum useful pulse width resolvable by logic gates.

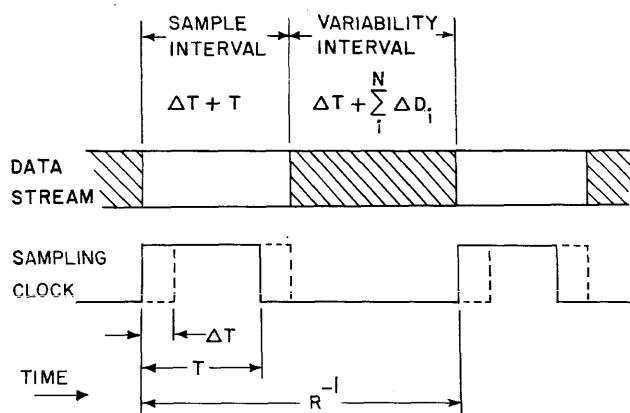


Figure 3—Continuous data stream

Figure 3. The sample interval is a function of the fixed clock-pulse width, defined to be T , and clock skew (ΔT). Skew ΔT is the total range of variability in arrival time of the clock pulse, as observed at the clocked latch. The clock could be present as late as time $T + \Delta T$, therefore the sample interval width is:

$$\text{Sample} = T + \Delta T \quad (4)$$

The variability interval is defined as the time interval over which one discrete data arrival occurs, but whose exact arrival time cannot be predicted, and half the time not observable because it is indistinguishable from the preceding state. The variability interval, observed at the receiving latch input, is a function of a composite combinatorial path leading from the sending latch clock. The variability additively builds up over path delay (P) through a pipeline section composed of N logic circuit levels, each with level delay D_i , such that

$$\begin{aligned}
 \text{Variability} &= (\Delta T + P_{\max}) - P_{\min} \\
 &= (\Delta T + \sum_{i=1}^N \max D_i) - \sum_{i=1}^N \min D_i \\
 &= \Delta T + \sum_{i=1}^N \widehat{D}_i - \sum_{i=1}^N \check{D}_i \\
 &= \Delta T + \sum_{i=1}^N (\widehat{D}_i - \check{D}_i) \\
 &= \Delta T + \sum_{i=1}^N \Delta D_i
 \end{aligned}$$

where $\Delta D_i = \widehat{D}_i - \check{D}_i$ (5)

Combining the results of (4) and (5), the maximum clock rate R is:

$$R = 1 / (\text{Sample} + \text{Variability}) \quad \text{cycles/second}$$

$$R = 1 / (T + 2 \Delta T + \sum_{i=1}^N \Delta D_i) \quad \text{cycles/second} \quad (6)$$

In practice, T can be made to approach one circuit delay for a one-circuit delay type of latch, Figure 4, and subnanosecond circuit delays are possible.¹⁷ The latch is the basic storage cell used in the sample and store operation. Based on past work⁵ reasonable values of ΔT are 10 percent/ R .

The most significant conclusion is that a summation of the circuit delay difference at each logic level constitutes the principle term to be minimized for higher rates. The effects of finite signal propagation times are thus reducible to arbitrarily small consequences, since in theory delay could be added to fast paths to minimize delay differences. Stating the conclusion more abstractly, the max-rate philosophy of design advocates minimizing differences in naturally occurring absolute quantities, whereas the classical approach to logic design has always been concerned with minimizing the absolute quantities themselves. Each philosophy will

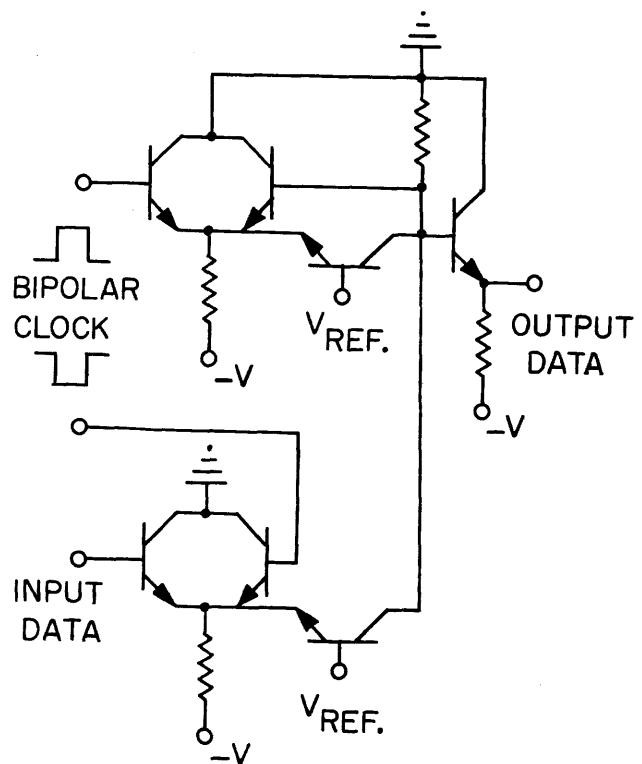


Figure 4—One-delay sampling latch

offer special advantages depending upon technology, systems goals, and acceptable design constraints.

Data rates and logic-path bandwidth

Clock pulse width depends primarily on circuit delay; however, variability is dependent on path bandwidths, which suffer from losses, mismatches, and loading. Theoretical data rates could approach the Nyquist rate, $2f$, where f is the abrupt cutoff frequency of an ideal low pass filter, approximated by the logic.¹⁴ The minimum width pulse or bit equals one Nyquist interval, or $1/2f$. In the environment a more practical definition of the Nyquist rate is the maximum signaling rate at which intersymbol interference vanishes. This implies full swings, without compromise of noise margins. In practice the Nyquist rate for a string of gates must be lower than that for a single gate. This results from pulse width variation that is not Gaussian in steep-cut-off bandwidth limited systems. These and circuit relaxation effects tend to hasten bit interference, especially for the isolated 1 or 0 bit cases, and could lead to smearing and pulse pinch-off for marginal rates.

In synchronous systems, where variability is accounted for by retiming, the ideal rate would probably never exceed half the Nyquist rate of logic gates. If repetitive clock pulses are furnished by a logic gate, the pulse width could equal one Nyquist interval, but pulses would have to be separated by one Nyquist interval. Under these assumptions the maximum realizable rate is:

$$R = 1/[2(T + \Delta T)] \quad \text{cycles/second} \quad (7)$$

where:

$\pm \Delta T/2$ = Random jitter for the limiting clock case.

$|T + 2\Delta T|$ = Sample interval magnitude

$|T|$ = Variability interval magnitude

The variability interval is taken to be one Nyquist interval for this special case, and would prove too severe for ordinary design purposes. Instead, as provided for in (6), an arbitrarily large variability interval may be accommodated by a corresponding rate reduction. It follows also that (7) establishes an upper bound for (6), assuming the pulse portion of the clock system is implemented with logic gates.

In practice designers empirically arrive at path-bandwidth and analytic approximations, as no present mode is available at this level of complexity. In the past, design relationships have been derived for each new tech-

nology and associated effects of loading, transmission line phenomena, and minimum useful pulse width. In addition to past measures of circuit delay, it is important that future max-rate technology be given additional definitions of performance. Of considerable interest are the Nyquist rates for worst-case bit patterns, taken for a continuum of input transition times. This characterization could null out package delay influences that otherwise appear as additive delay in present measurements. The lead and package delay should be treated as wire delay which delays computation, but need not cause a reduction in rate. By similar reasoning a clock pulse width, on the order of a Nyquist interval, should not be widened to account for finite lead-package delay encountered while entering a latch of Lilliputian dimensions compared to the package. A second type of performance characterization should be statistical measures of delay variability to account for production spreads, variable input transition times, and asymmetry in circuit switching characteristics.

Minimizing logic level variability

The throughput data rate approaches the maximum realizable clocked rate (7) when in (6):

$$\sum_{i=1}^N \Delta D_i \rightarrow |T| \quad (8)$$

It is significant that the expression (8) to be minimized is a summation of artificial differences, as contrasted with natural barriers such as signal propagation velocity and combinatorial gate delay through $N-1$ levels of logic between clocks.

A straightforward variability minimization algorithm is needed if several complex considerations are to be avoided. Statistical averaging through several levels of gating must not be used, as this places complex dependencies on all gates in a section. This is hazardous because in a regular m by n gate array there are m^n paths possible, and these paths could merge $m!n!/(m-2)!(2)!$ ways. Any of these mergers, because of delay differences, could have relative bit offsets that might cause symbol interference at an output. One algorithm that avoids this complexity is to make each logic level independent from the others. This can be implemented by permitting designers to reduce $\Delta D_i = \bar{D}_i - \check{D}_i$ on a level by level basis by any means available, but requiring that ultimately a max D_i and min D_i be specified for every level. Designers could tailor each level to achieve maximum speed, or design could be standardized about a few restricted cases to cover all situations.

Wire and loading delay is included at each level. Only the uncontrolled differences in the circuit-line-loading delay triplet need contribute to ΔD_i . Even here, longer path routing or more loading may be used to slow down fast lightly loaded gates. Increases in transition times due to loading are accounted for either in increased transmission line delay or increased delay of driven gates. Parallel line delays such as the six nanosecond delays shown in Figure 5 would not affect the maximum rate as calculated in (6). In fact (6), along with the Figure 5 example data shown in Table I, predicts a 125 MHz clock rate. By contrast, 42 MHz results if a summation of maximum delays along with clock skew is permitted to determine the period of the clock.

Clock phasing

In order to insure sampling from the nonchanging portions of bit intervals each receiving station clock

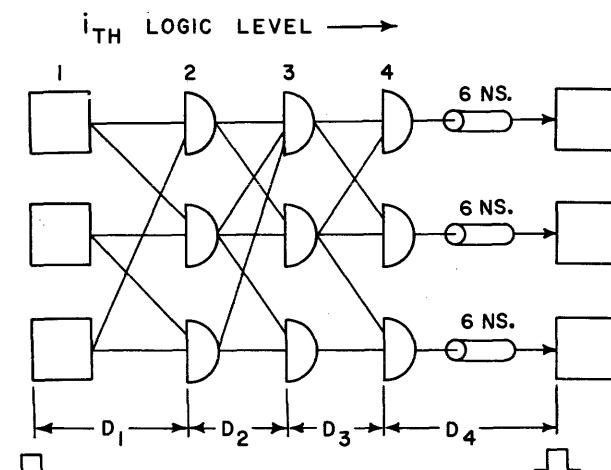


Figure 5—Representative pipeline section

Table I—Example delay data

i	1	2	3	4
\hat{D}_i NS.	4.1	5.9	4.2	3.0 + 6.0
\check{D}_i NS.	3.7	4.8	3.6	2.7 + 6.0
ΔD_i NS.	0.4	1.1	0.6	0.3 + 0
CLOCK WIDTH	$T = 4.0$ NS.			
CLOCK SKEW	$\Delta T = 0.8$ NS.			

must be offset from the sending station clock an amount:

$$\text{Clock offset} = (\Delta T + \sum_{i=1}^N \hat{D}_i) \text{ Modulo } 1/R \quad (9)$$

This resembles a spatially rippled clock which has been mentioned.¹⁸ This requirement could be met by distributing a global time reference, treated as an independent machine,¹⁹ about a 50,000 gate system to within phase differences of say ± 0.2 ns. Next a delay element similar to Figure 6 could be used at each place where a phase is needed, typically for each 150-300 gate section. A stored or wired address would establish the phase to be supplied.

Since all phases were assumed to exist in (6) and actually only p discrete phases are available some reduction in rate becomes necessary. The worst-case rate reduction need never exceed a decrease of 100 percent/p. In reality one would use the a priori knowledge concerning phasing, and constrain the variability interval, thus the reduction in rate is more a theoretical entity. Lastly, the most attractive approach toward realizing some systems is to constrain design such that (9) meets the zero offset case for all sections. This results in a single phase clock, and the requirement for multiple phases no longer exists.

Troubleshooting a max-rate system

Troubleshooting and stepping a maximum-rate system has some new aspects. At any time a dynamic path may contain more bits than storage latches such that storage is not available for the excess bits, or:

$$\text{Excess bits} = R (\Delta T + \sum_{i=1}^N \hat{D}_i) - 1 \quad (10)$$

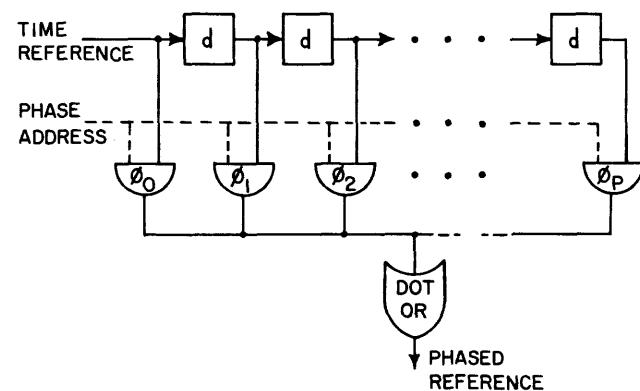


Figure 6—Delay with gated phases

This bears similarity to a delay line. The point to be remembered is that max-rate machines possess the complete personality of a single phase machine, with no excess bits, operating at a lower rate determined by the max-delay section in the system. Also, to be race free in this mode a requirement must be met such that:

$$\sum_{i=1}^N \bar{D}_i \geq |T| \quad (11)$$

Condition (11) merely acknowledges that one could perhaps be using a conservative sampling pulse width, much greater than delay through fast paths between some latches.

CONCLUSIONS

It is possible for max-rate pipeline machines to operate at high rates determined by path differences, rather than the conventional maximum delay. The results apply to any technology, but would prove most useful when signal propagation delays approach or exceed circuit delay. In this case velocity of propagating signals need not limit rates if paths are equalized. The approach described would permit increased performance in present systems environments, and would pave the way for entry into the subnanosecond regime where relatively long transmission lines might exist between gate arrays.

REFERENCES

- 1 M J FLYNN
Very high-speed computing systems
IEEE Proc Vol 54 No 12 December 1966
- 2 R A ASCHENBRENNER M J FLYNN
G A ROBINSON
Intrinsic multiprocessing
Proc S J C C 1967
- 3 R A ASCHENBRENNER
Time-phased parallelism
Proc National Electronics Conference 1967
- 4 G M AMDAHL
Validity of the single processor approach to achieving large scale computing capabilities
Computer Design December 1967 and
Proc S J C C 1967
- 5 L W COTTEN
- 6 S F ANDERSON J G EARLE
R E GOLDSCHMIDT D M POWERS
The IBM System/360 Model 91: floating point execution unit
IBM Journal of R & D January 1967
- 7 D N SENZIG
Observations on high-performance machines
Proc F J C C 1967
- 8 D L SLOTNICK
Unconventional systems
Computer Design December 1967 and
Proc S J C C 1967
- 9 W D LEWIS
Microwave logic
Proc of International Symposium on the Theory of Switching 1957
- 10 M O PALEY
LSI and the large computer systems designer
International Solid State Circuits Conference 1968
- 11 A R STRUBE
LSI for high-performance logic applications
International Electron Devices Meeting 1967
- 12 H H LOOMIS JR
The maximum-rate accumulator
IEEE Transactions on Electronic Computers Vol EC-15 No 4 August 1966
- 13 J M WIER
Digital data communications techniques
Proc of IRE Vol 49 January-March 1961
- 14 W R BENNETT J R DAVEY
Data transmission
McGraw-Hill Inc N Y 1965
- 15 D N ARDEN
Delayed logic and finite state machines
Proc of AIEE Symposium on Switching Theory and Logical Design September 1961
- 16 H H LOOMIS JR
Completeness of sets of delayed-logic devices
IEEE Transactions on Electronic Computers Vol EC-14 No 2 April 1965
- 17 D H CHUNG
The design considerations of picosecond integrated switching circuits
SWIEECO 1966
- 18 B ELSPAS J GOLDBERG R C MINNICK
R A SHORT H S STONE
Investigation of propagation-limited computer networks
Final Report Air Force Contract AF19(628)-2902
Stanford Research Institute Project 4523 1964
- 19 J HARTMANIS
Maximal autonomous clocks of sequential machines
IEEE Transactions on Electronics Computers Vol EC-11 No 1 February 1962