Clock Interface

10

This chapter describes the clock signals ("clocks") used in the R4000 processor and the processor status reporting mechanism.

The subject matter includes basic system clocks, system timing parameters, connecting clocks to a phase-locked system, connecting clocks to a system without phase locking, and processor status outputs.

10.1 Signal Terminology

The following terminology is used in this chapter (and book) when describing signals:

- *Rising edge* indicates a low-to-high transition.
- *Falling edge* indicates a high-to-low transition.
- *Clock-to-Q delay* is the amount of time it takes for a signal to move from the input of a device (*clock*) to the output of the device (*Q*).

Figures 10-1 and 10-2 illustrate these terms.

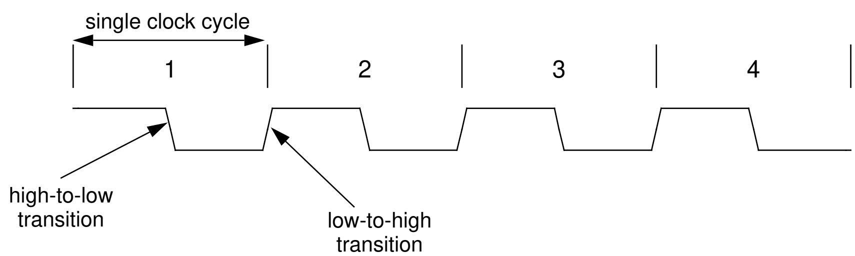

Figure 10-1 Signal Transitions

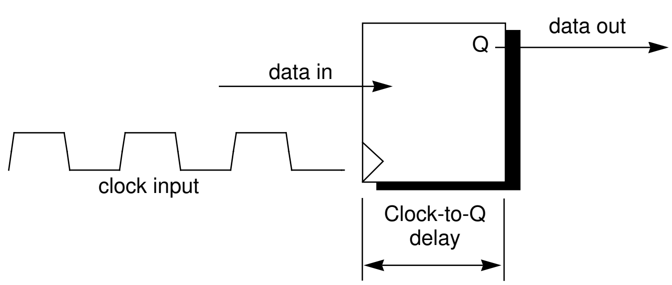

Figure 10-2 Clock-to-Q Delay

10.2 Basic System Clocks

The various clock signals used in the R4000 processor are described below, starting with **MasterClock**, upon which the processor bases all internal and external clocking.

MasterClock

The processor bases all internal and external clocking on the single **MasterClock** input signal. The processor generates the clock output signal, **MasterOut**, at the same frequency as **MasterClock** and aligns **MasterOut** with **MasterClock**, if **SyncIn** is connected to **SyncOut**.

MasterOut

The processor generates the clock output signal, **MasterOut**, at the same frequency as **MasterClock** and aligns **MasterOut** with **MasterClock**, if **SyncIn** is connected to **SyncOut**. **MasterOut** clocks external logic, such as the reset logic.

SyncIn/SyncOut

The processor generates **SyncOut** at the same frequency as **MasterClock** and aligns **SyncIn** with **MasterClock**.

SyncOut must be connected to **SyncIn** either directly, or through an external buffer. The processor can compensate for both output driver and input buffer delays (and, when necessary, delay caused by an external buffer) when aligning **SyncIn** with **MasterClock**. Figure 10-7 gives an illustration of **SyncOut** connected to **SyncIn** through an external buffer.

PClock

The processor generates an internal clock, **PClock**, at twice the frequency of **MasterClock** and precisely aligns every other rising edge of **PClock** with the rising edge of **MasterClock**.

All internal registers and latches use **PClock**.

SClock

The R4000 processor divides **PClock** by 2, 3, or 4 (as programmed at boot-mode initialization) to generate the internal clock signal, **SClock**. The R4400 processor divides **PClock** by 2, 3, 4, 6 or 8 (as programmed at boot-mode initialization) to generate **SClock**. The processor uses **SClock** to sample data at the system interface and to clock data into the processor system interface output registers.

The first rising edge of **SClock**, after **ColdReset*** is deasserted, is aligned with the first rising edge of **MasterClock**.

TClock

TClock (transmit clock) clocks the output registers of an external agent,† and can be a global system clock for any other logic in the external agent.

TClock is the same frequency as **SClock**. When **SyncIn** is shorted to **SyncOut**, the edges of **TClock** align precisely with the edges of **SClock** and **MasterClock**.

When a delay is added between **SyncIn** and **SyncOut**, the **TClock** at the pins leads **SClock** (and thus **MasterClock**) by the same amount of delay.

If the delay between **SyncIn** and **SyncOut** is matched to an external delay between **TClock** at the processor and **TClock** at the external logic, the **TClock** at the external logic aligns to **SClock** and **MasterClock**.

RClock

The external agent uses **RClock** (receive clock) to clock its input registers. The processor generates **RClock** at the same frequency as **TClock**, but **RClock** always leads **TClock** and **SClock** by 25 percent of **SClock** cycle time. The relationship between **RClock** and **TClock** is independent of the delay between **SyncIn** and **SyncOut**.

PClock-to-SClock Division

Figure 10-3 shows the clocks for a **PClock**-to-**SClock** division by 2; Figure 10-4 shows the clocks for a **PClock**-to-**SClock** division by 4.

† *External agent* is defined in Chapter 12.

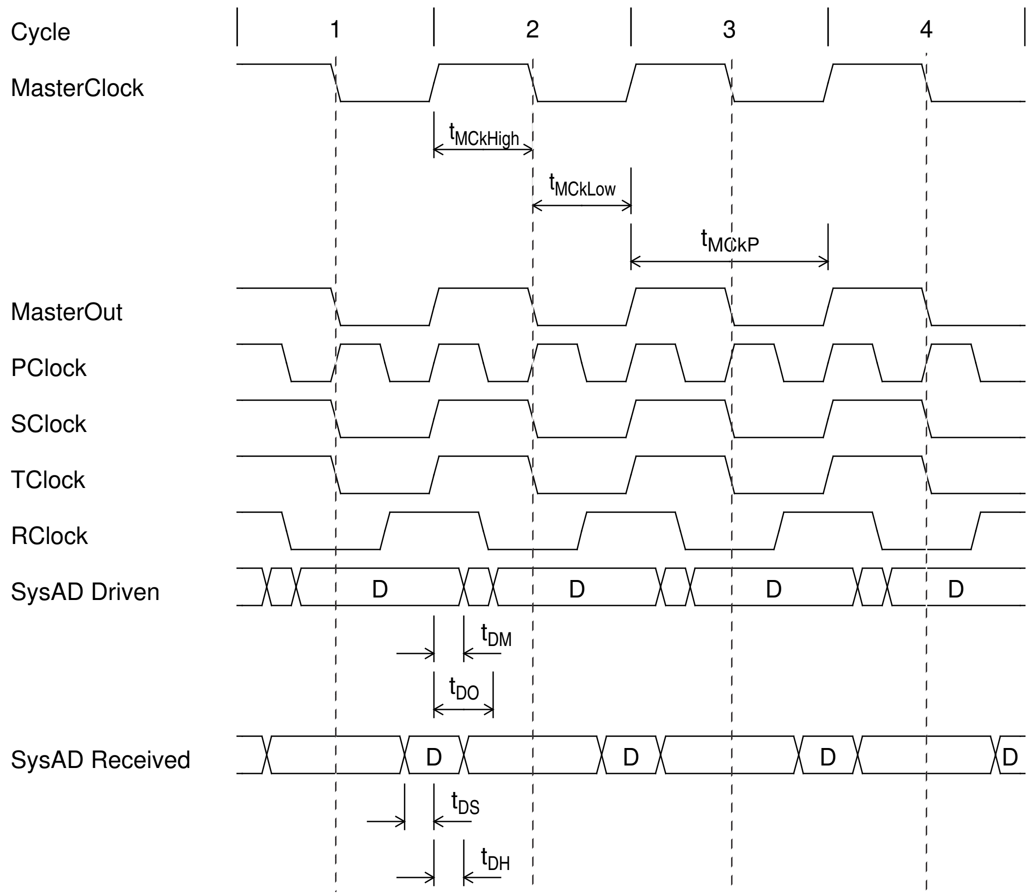

Figure 10-3 Processor Clocks, PClock-*to*-SClock *Division by 2*

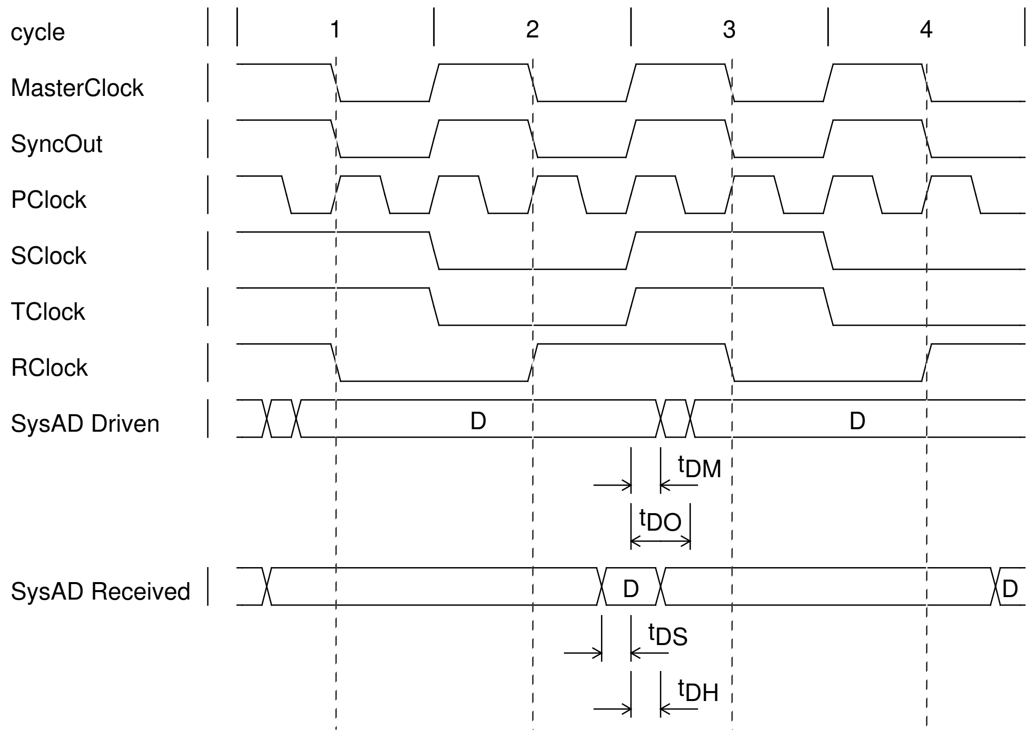

Figure 10-4 Processor Clocks, PClock-*to*-SClock *Division by 4*

10.3 System Timing Parameters

As shown in Figures 10-3 and 10-4, data provided to the processor must be stable a minimum of t_{DS} nanoseconds (ns) before the rising edge of **SClock** and be held valid for a minimum of t_{DH} ns after the rising edge of **SClock**.

Alignment to SClock

Processor data becomes stable a minimum of t_{DM} ns and a maximum of t_{DO} ns after the rising edge of **SClock**. This drive-time is the sum of the maximum delay through the processor output drivers together with the maximum clock-to-Q delay of the processor output registers.

Alignment to MasterClock

Certain processor inputs (specifically **VCCOk**, **ColdReset***, and **Reset***) are sampled based on **MasterClock**, while others (specifically, **Status(7:0)**) are output based on **MasterClock**. The same setup, hold, and drive-off parameters, t_{DS}, t_{DH}, t_{DM}, and t_{DO}, shown in Figures 10-3 and 10-4, apply to these inputs and outputs, but they are measured by **MasterClock** instead of **SClock**.

Phase-Locked Loop (PLL)

The processor aligns **SyncOut**, **PClock**, **SClock**, **TClock**, and **RClock** with internal phase-locked loop (PLL) circuits that generate aligned clocks based on **SyncOut**/**SyncIn**. By their nature, PLL circuits are only capable of generating aligned clocks for **MasterClock** frequencies within a limited range.

Clocks generated using PLL circuits contain some inherent inaccuracy, or *jitter*; a clock aligned with **MasterClock** by the PLL can lead or trail **MasterClock** by as much as the related maximum jitter allowed by the individual vendor.

10.4 Connecting Clocks to a Phase-Locked System

When the processor is used in a phase-locked system, the external agent must phase lock its operation to a common **MasterClock**. In such a system, the delivery of data and data sampling have common characteristics, even if the components have different delay values. For example, *transmission time* (the amount of time a signal takes to move from one component to another along a trace on the board) between any two components A and B of a phase-locked system can be calculated from the following equation:

Transmission Time = (**SClock** period) – (t_{DO} for A) – (t_{DS} for B) –
(Clock Jitter for A Max) – (Clock Jitter for B Max)

Figure 10-5 shows a block-level diagram of a phase-locked system using the R4000 processor.

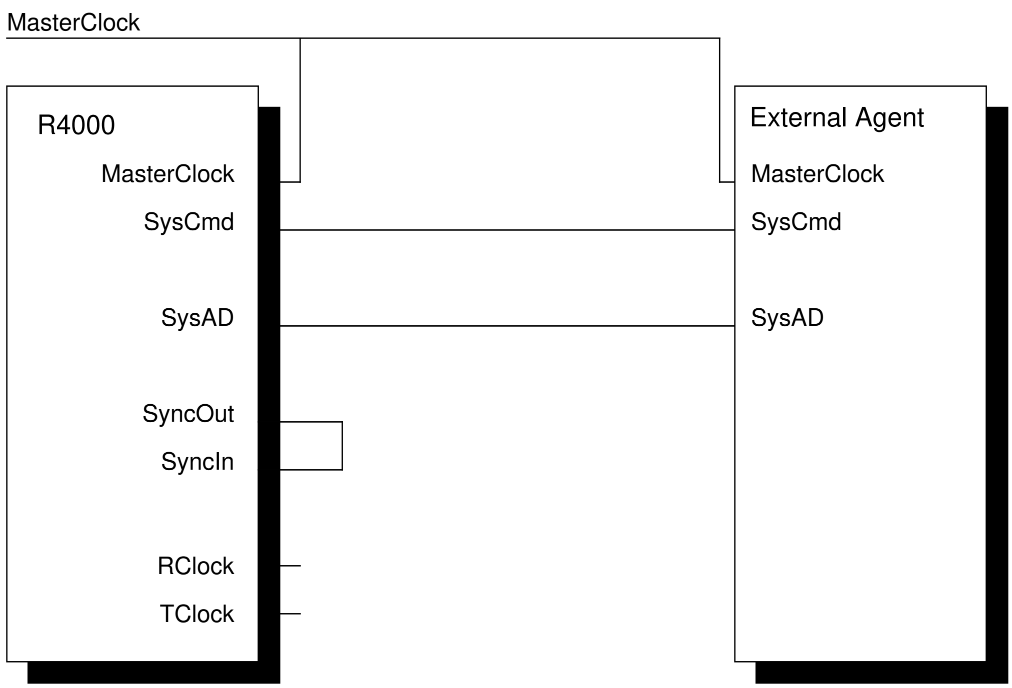

Figure 10-5 R4000 Processor Phase-Locked System

10.5 Connecting Clocks to a System without Phase Locking

When the R4000 processor is used in a system in which the external agent cannot lock its phase to a common **MasterClock**, the output clocks **RClock** and **TClock** can clock the remainder of the system. Two clocking methodologies are described in this section: connecting to a gate-array device or connecting to discrete CMOS logic devices.

Connecting to a Gate-Array Device

When connecting to a gate-array device, both **RClock** and **TClock** are used within the gate-array. The gate array internally buffers **RClock** and uses this buffered version to clock registers that sample processor outputs.

These sampling registers should be immediately followed by staging registers clocked by an internally buffered version of **TClock**. This buffered version of **TClock** should be the global system clock for the logic inside the gate array and the clock for all registers that drive processor inputs. Figure 10-6 is a block diagram of this circuit.

Staging registers place a constraint on the sum of the clock-to-Q delay of the sample registers and the setup time of the staging registers inside the gate arrays, as shown in the following equation:

$$\left.\begin{array}{l}\text{Clock-to-Q Delay + Setup of Staging Register} \\ \quad - \text{(Maximum Clock Jitter for RClock)} \\ \quad - \text{(Maximum Delay Mismatch for Internal Clock} \\ \qquad\qquad \text{Buffers on RClock and TClock)}\end{array}\right\} \quad 0.25 \text{ (RClock period)}$$

Figure 10-6 is a block diagram of a system without phase lock, using the R4000 processor with an external agent implemented as a gate array.

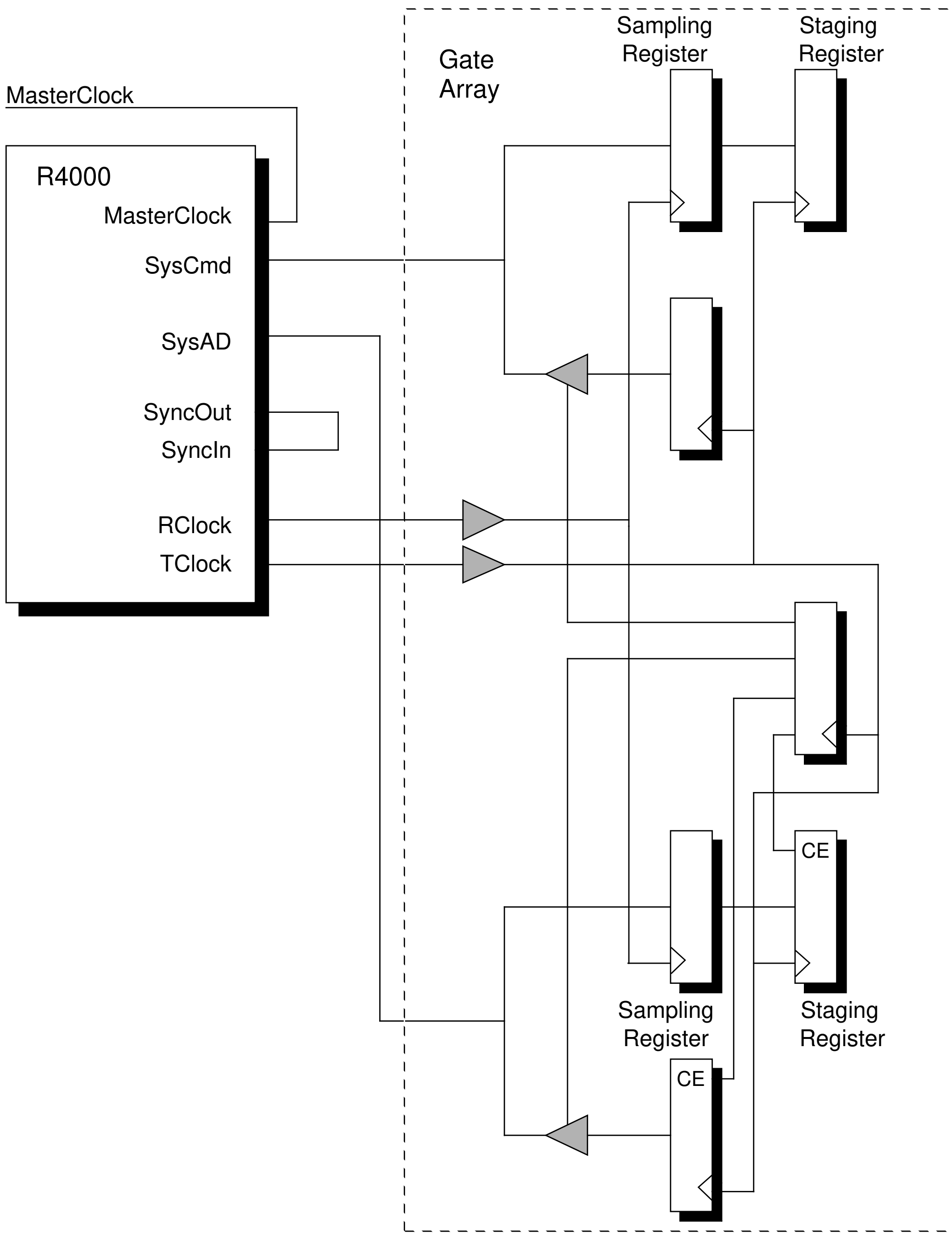

Figure 10-6 Gate-Array System without Phase Lock, using the R4000 Processor

In a system without phase lock, the transmission time for a signal *from* the processor *to* an external agent composed of gate arrays can be calculated from the following equation:

Transmission Time = (75 percent of **TClock** period) – (t_{DO} for R4000)
+ (Minimum External Clock Buffer Delay)
– (External Sample Register Setup Time)
– (Maximum Clock Jitter for R4000 Internal Clocks)
– (Maximum Clock Jitter for **RClock**)

The transmission time for a signal *from* an external agent composed of gate arrays *to* the processor in a system without phase lock can be calculated from the following equation:

Transmission Time = (**TClock** period) – (t_{DS} for R4000)
– (Maximum External Clock Buffer Delay)
– (Maximum External Output Register Clock-to-Q Delay)
– (Maximum Clock Jitter for **TClock**)
– (Maximum Clock Jitter for R4000 Internal Clocks)

Connecting to a CMOS Logic System

The processor uses matched delay clock buffers to generate aligned clocks to external CMOS logic. A matched delay clock buffer is inserted in the **SyncOut**/**SyncIn** alignment path of the processor, skewing **SyncOut**, **MasterOut**, **RClock**, and **TClock** to lead **MasterClock** by the buffer delay amount, while leaving **PClock** aligned with **MasterClock**.

The remaining matched delay clock buffers are available to generate a buffered version of **TClock** aligned with **MasterClock**. Alignment error of this buffered **TClock** is the sum of the maximum delay mismatch of the matched delay clock buffers, and the maximum clock jitter of **TClock.**

As the global system clock for the discrete logic that forms the external agent, the buffered version of **TClock** clocks registers that sample processor outputs, as well as clocking the registers that drive the processor inputs.

The transmission time for a signal from the processor to an external agent composed of discrete CMOS logic devices can be calculated from the following equation:

Transmission Time = (**TClock** period) – (t_{DO} for R4000)

– (External Sample Register Setup Time)

– (Maximum External Clock Buffer Delay Mismatch)

– (Maximum Clock Jitter for R4000 Internal Clocks)

– (Maximum Clock Jitter for **TClock**)

Figure 10-7 is a block diagram of a system without phase lock, employing the R4000 processor and an external agent composed of both a gate array and discrete CMOS logic devices.

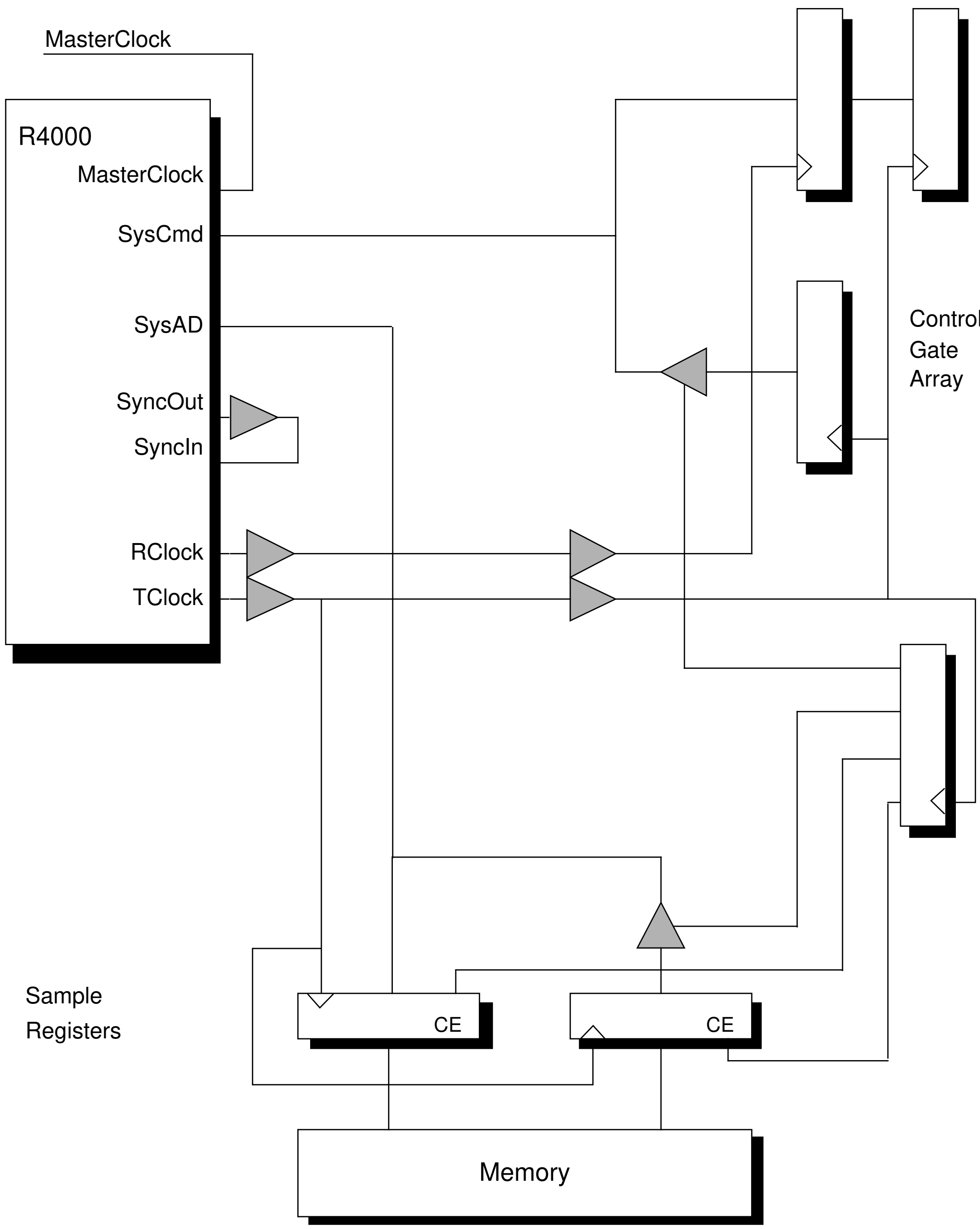

Figure 10-7 Gate Array and CMOS System without Phase Lock, using the R4000 Processor

The transmission time for a signal from an external agent composed of discrete CMOS logic devices can be calculated from the following equation:

Transmission Time = (**TClock** period) – (t_{DS} for R4000)
– (Maximum External Output Register Clock-to-Q Delay)
– (Maximum External Clock Buffer Delay Mismatch)
– (Maximum Clock Jitter for R4000 Internal Clocks)
– (Maximum Clock Jitter for **TClock**)

In this clocking methodology, the hold time of data driven from the processor to an external sampling register is a critical parameter. To guarantee hold time, the minimum output delay of the processor, t_{DM}, must be greater than the sum of:

minimum hold time for the external sampling register
+ maximum clock jitter for R4000 internal clocks
+ maximum clock jitter for **TClock**
+ maximum delay mismatch of the external clock buffers

10.6 Processor Status Outputs

The R4400 processor provides eight status outputs, **Status(7:0)**, aligned with each rising edge of **MasterClock**. At time T (the first PCycle of **MasterClock** when status is examined) these status outputs indicate whether the machine was running or stalled during the previous T-2 and T-3 PCycles, as follows:

- If the machine was stalled during the T-2 or T-3 PCycles, the status outputs indicate the type of stall which occurred (listed in Table 10-1).
- If the machine was running during the T-2 or T-3 PCycles, the status outputs describe the type of instruction which occupied the WriteBack pipeline stage during the T-2 or T-3 PCycles, and which was successfully completed (listed in Table 10-1).
- The status outputs also indicate if an instruction in the T-2 or T-3 PCycle was killed, and if so, states the cause (listed in Table 10-1.

The **Status(7:0)** bits are treated as two fields, as follows:

- The **Status(7:4)** field indicates the internal status of the processor during PCycle T-3.
- The **Status(3:0)** bits indicate the internal status of the processor during the PCycle T-2.

Table 10-1 shows the encoding of processor's status for pins **Status(7:4)** or **Status(3:0)**.

Table 10-1 Encoding of R4400 Processor Internal State by Status(7:4) *or* Status(3:0)

Status(7:4) or Status(3:0)	Cycle	Processor Internal Status
0	Run cycle	Other integer instruction (not load/store/conditional branch. Includes ERET and Jump instructions.)
1	Run cycle	Load
2	Run cycle	Untaken conditional branch
3	Run cycle	Taken conditional branch
4	Run cycle	Store
5	Reserved	
6	Stall cycle	MP stall
7	Run cycle	Integer instruction killed by slip
8	Stall cycle	Other stall type
9	Stall cycle	Primary instruction cache stall
a	Stall cycle	Primary data cache stall
b	Stall cycle	Secondary cache stall
c	Run cycle	Other floating-point instruction (not load, store, or conditional branch)
d	Run cycle	Instruction killed by branch, jump, or ERET
e	Run cycle	Instruction killed by exception
f	Run cycle	Floating-point instruction killed by slip