2.1 General Design Considerations

Principal Author: John Poulton

Microelectronic Systems Laboratory

The University of North Carolina at Chapel Hill
Department of Computer Science
New West Hall 035 A
Chapel Hill, N.C. 27514

Table of Contents

2.1. General Design Considerations

Introduction

This document outlines a general method for dealing with signals and inter-module communications in the Pixel-planes Graphic Engine.

Section 2.1.1 describes conventions and specifications for the electrical connections between modules in all Pixel-planes chip designs. A signal naming convention is first described, that includes a method for 'strong typing' signals in an attempt to ensure the correct interconnection of modules with respect to signal polarity, timing, etc. The section then specifies the rules for module interconnection based on signal types.

Section 2.1.2 specifies the schematic symbol conventions to be used in the Pixel-planes project.

Section 2.2.3 describes the canonical method for interfacing a chip to its external environment. It specifies

- How the single-phase clocks used on external busses are converted to the 2-phase non-overlapping clocks used on-chip.
- The timing constraints needed to guarantee that external bus signals satisfy the timing requirements of on-chip circuits for chip inputs.
- The timing characteristics of chip outputs and the timing constraints for correctly latching these signals.

2.1.1. Signal Conventions

2.1.1.1. Signal Naming

The following rules define the signal naming conventions for the Pixel-planes graphic engine design:

(1) Signal names shall not exceed five characters (exclusive of suffixes and signal type–see below).

(2) Signal names are written in lower case with upper case letters at word boundaries (e.g., the signal for 'write enable' would become **WrtEn**).

(3) Groups of closely-related signals are specified by an ISP-like array notation <num:num>. For example, the notation **Dat<0:7>** expands to the eight signals **Dat0, Dat1, Dat2 ... Dat7**.

(4) The signal names **Vdd**, **vdd**, **VHot**, **Vss**, **vss**, and **GND** are reserved for the power rails and are not typed. Although **Vss** and **GND** are exactly equivalent, as are **vdd** and **Vdd**, the two sets of names are needed to suit the requirements of various design software tools.

2.1.1.2. On-Chip Signal Typing

The signal typing described in this section specifies timing properties of a signal as referred to the internal clocks of custom chips in the Pixel-planes system. These chips include those in the **Image Generation Controller** and in the **Engine**.

Two kinds of signal types are defined: fundamental types, which are the only types allowed at the inputs and outputs of a design module, and ancillary types, used only to described signals internal to lowest-level design modules. Each signal type implies some information about the timing and voltage characteristics of that signal.

Important Note: The voltage/timing characteristics specified for each signal type are NOT automatically guaranteed to obtain for any given circuit construct. Rather, these specifications should be taken as directives to the designer, who must guarantee that each module meets the specification for the signal type associated with each input and output of that module. Specifically, these specifications must be met when the

module is instantiated in its intended environment.

Timing characteristics refer to specific points on the rising and trailing edge of waveforms, as defined in Figure 2.1.1.

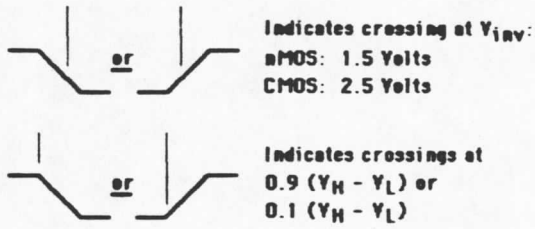


Figure 2.1.1: Timing diagram conventions.

The 2-phase strategy is employed, typically, within the boundaries of a chip. External to the chip, it is usually more convenient to synchronize events on a single phase clock; signal typing for single-phase clocking is described in section 2.1.1.3.

Fundamental Signal Types

A hierarchy of signal types exists in this scheme. Clocks, the reference for other type definitions, are at the top of this hierarchy. Below this level are two signal types, **S** and **E**, that are the only types allowed as data/control inputs and outputs of design modules. At the bottom of the hierarchy are several types used to describe intra-module signals; these types are are described in the following section.

Clocks

Clocks are the fundamental reference for other type definitions. In the Pixel-planes 'smart frame buffer' design, two clocks control the timing of two separate processes: the **Ph** clock times image generation circuitry (Image Generation Controller--section 2.3, and image-generation port of the Pxpl4 memory chip--section 2.4), while the **Px** clock times video data scan-out (Video Controller--section 2.5 and video-data port of Pxpl4 memory chip). The precise timing and voltage characteristics of clocks are fully specified in section 2.1.3.1. For purposes of signal typing, assume that clocks toggle between the supply rails (CMOS), or between **GND** and **VHot** (nMOS).

Qualified clocks are generated from the logical AND between a signal, stable on a phase, and that phase of the clock; the canonical scheme for generating qualified clocks is described in 2.1.3.1. A qualified clock is completely named by joining the signal name to a type field that is just the name of the related clock. For example, a memory word line asserted on **Ph2** might have the signal name **WdHPh2H**. Waveforms and timing constraints for clocks and qualified clocks are shown in Figure 2.1.2.

Note: In CMOS, both polarities of each clock may be required, so that the signals **Ph1H, Ph2H, Ph1L, Ph2L** for image-generation timing must all be defined. In nMOS, generally only one clock polarity is required; if this is true universally through a design, it is permissible to shorten the clock names and qualified-clock type fields to **Ph1** and **Ph2**.

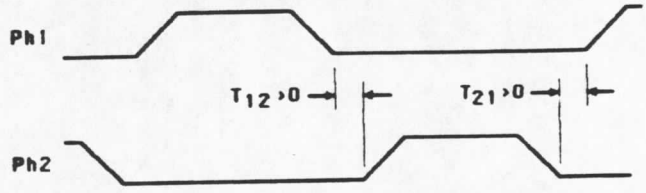


Figure 2.1.2: Clock timing constraints.

For the two signal types used to communicate data and control between design modules, a signal is fully named by joining a **signal name** with a four-character **signal type**. The four characters of the signal type comprise four single-character fields, whose functions are:

Field	Allowed Char's	Function
1	**H,L**	Indicates active-HI or active-LO
2	**S,E**	Timing characteristics--see below
3	**P,X**	Associated clock (**Ph** or **Px**)
4	**1,2**	Relevant clock phase

An example of a legal signal type is **HSX1**, indicating an active-HI signal, stable during **Px1**.

The two timing characteristics are:

S.

Stable on the indicated phase of the clock, but may change on the opposite phase. **S** outputs are pulled to one of the supply rails during the stable phase, and never go to a high-impedance state. Timing for the **S** type, together with a typical circuit is shown in Figure 2.1.3.

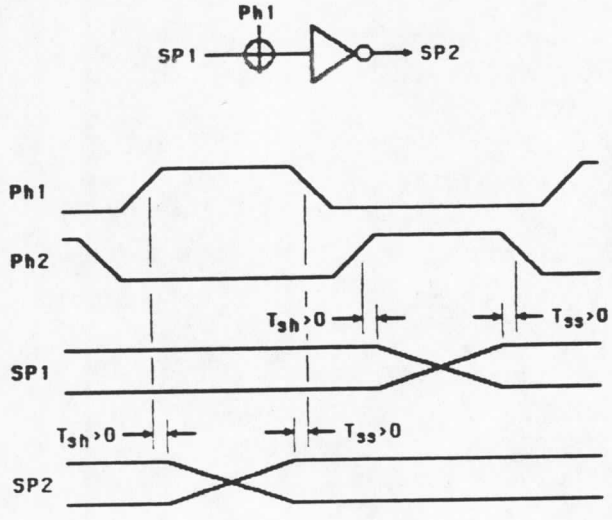
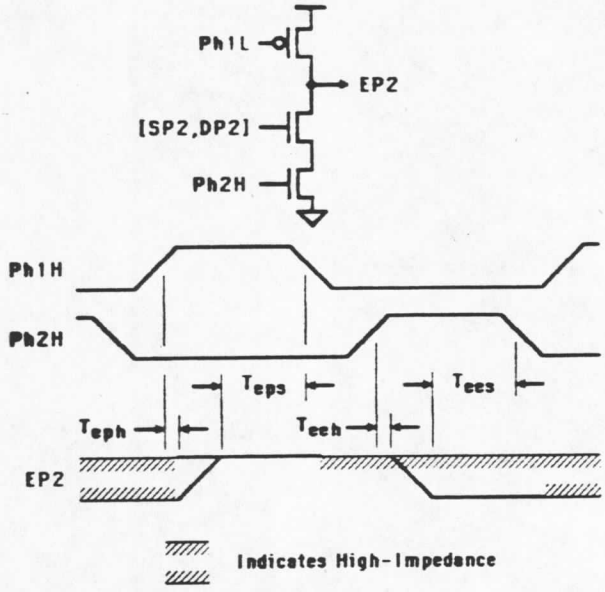


Figure 2.1.3: Typical circuitry and timing for S-type signals.

Note that the timing constraints for this type can be specified in a dimensionless way. The constraint on $\mathbf{T_{ss}}$ must be met at outputs of design modules *when they are instantiated in their electrical environment.*

E

Evaluated on the indicate phase. An output of type **EP2**, for example, is assumed to have been precharged to V_{dd} on **Ph1**; the pin is evaluated on **Ph2**, either leaving it precharged (and at high-impedance) or pulling it LO. In nMOS realizations of Pixel-planes circuits, the precharging signal is *always* a 'hot' clock, so that an **E**-type node can satisfy the requirement of being precharged to the V_{dd} rail. A typical circuit construction that generates an **E**-type signal and the timing properties for the type are shown in Figure 2.1.4.



Timing Constraints for E-Type Signals	
Constraint	Remarks
Teph > 0	Precharge hold time--signal must remain evaluated until after leading edge of clock
Teps > 0	Precharge setup time--signal must be fully precharged before trailing edge of clock
Teeh > 0	Evaluate hold time--signal must remain precharged until after leading edge of clock
Tees *	Evaluate setup time--see text.

Figure 2.1.4: Typical circuitry and timing for E-type signals.

Note that timing constraints for the precharge phase can be specified in a non-metric way. The timing constraints for the evaluation phase of these four dynamic signal types is more difficult to specify--a simple non-metric constraint isn't sufficient. To define the constraints, this document takes the view that the **E**-type signal is lower in the hierarchy than the **S**-type, and timing constraints for evaluation are referred to associated **S**-type nodes. To determine the constraints on a generator of an **E**-type signal, the designer must draw the smallest boundary around all circuitry whose internal nodes are of types **E** such that all signals entering or leaving through that boundary are of types **Clock** or **S**. The circuitry inside that boundary must be designed to satisfy the timing constraints for the **S**-type outputs emerging from the boundary.

Figure 2.1.5 shows an example of this rule.

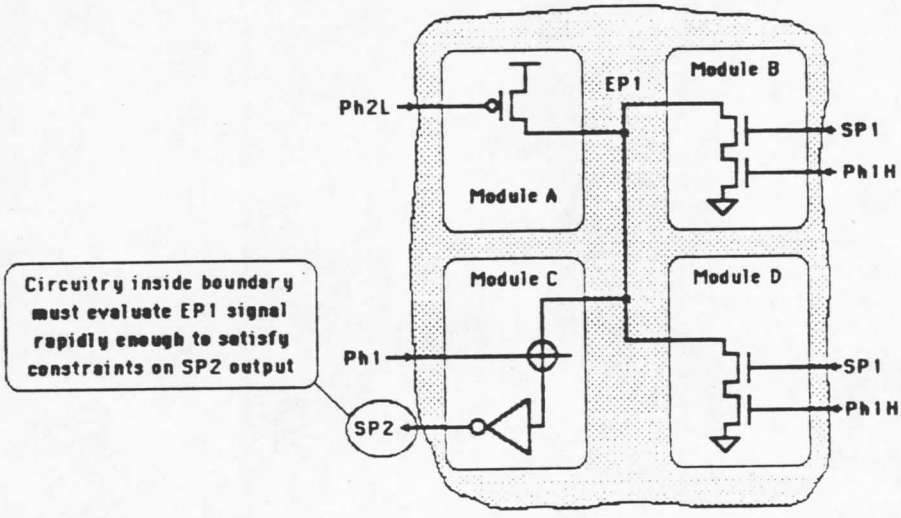


Figure 2.1.5: Determining timing constraints for E-type evaluation.

Ancillary Signal Types

For purposes of aiding the documentation of design modules, several ancillary signal types are defined.

C Clocked on the indicate phase. This class of signal describes the output nodes of pass-transistor networks whose control inputs are clocked on the indicate clock phase. In nMOS realizations of the Pixel-planes design, the gates of *all* clocked pass transistor networks are driven by 'hot' clocks, so that a C-type output is guaranteed to swing between the supply rails. A circuit example and waveforms for **C**-type signals are shown in Figure 2.1.6.

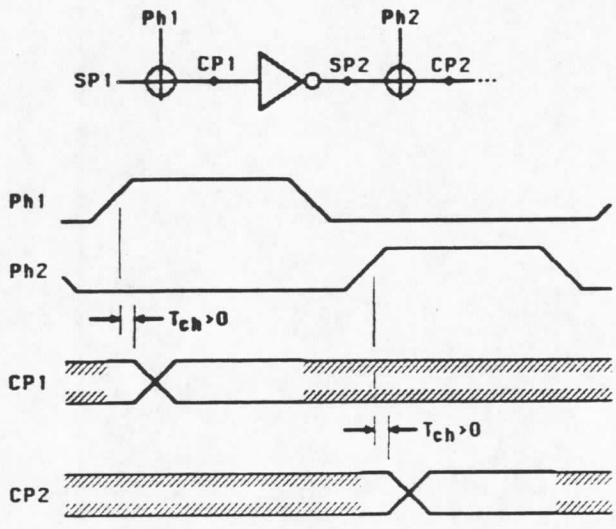


Figure 2.1.6: Typical circuitry and timing for C-type signals.

G Gated on the indicated phase. This signal type is defined to take care of circuits (such as pass-transistor logic networks) where one or more n-enhancement transistors are in the path between the typed node and **Vdd**. G-type signals have the same timing characteristics as **S** signals, but are not guaranteed to provide a HI output voltage greater than $V_{dd} - V_{th}$. Like **S** nodes, **G** nodes are never at high-impedance. Note that typical nMOS output pads utilizing enhancement pull-ups have **G** outputs according to this definition. In CMOS circuits, it is possible (but unlikely) to construct a circuit which inserts a threshold drop in the LO output voltage. Some examples of circuits that generate **G**-type signals are shown in Figure 2.1.7.

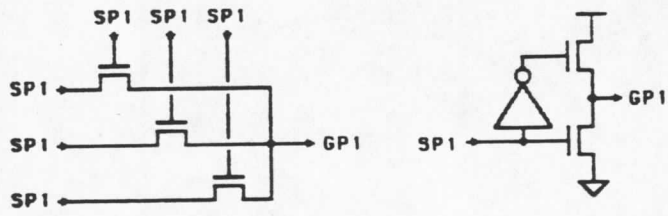


Figure 2.1.7: Circuit examples for G-type signals.

D Domino on the indicated phase. An output of type **DPn** is generated from a type **EPn** by inserting a level-restoring static logic element, typically an inverter. A pin of type **DP2** has the property that it is guaranteed to be LO at the end of **Ph1** and *may* go HI during **Ph2**. Domino outputs swing between the supply rails. If two or more stages of logic, whose outputs are type **DPn** are cascaded, all outputs start out LO at the beginning of **Phn**; if the output of the first stage then goes HI, it can cause the next stage's output to be evaluated HI, and so forth. Thus the name 'domino'. Figure 2.1.8 shows circuit examples, waveforms, and precharge timing constraints **D**-type signals.

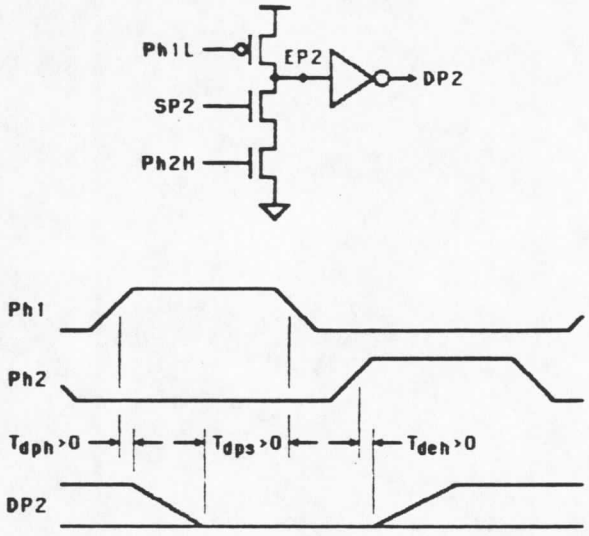


Figure 2.1.8: Circuitry and timing for D-type signals.

2.1.1.3. External Signal Typing

A signal typing scheme for signals external to chips is defined in this section. All signals in this typing scheme are assumed to toggle between standard TTL levels (logic LO=0.8 volts, logic Hl=2.0 volts). Timing specifications that refer to points on rising and falling edges that correspond to V_m=1.5 volts.

Four types are defined:

Clocks

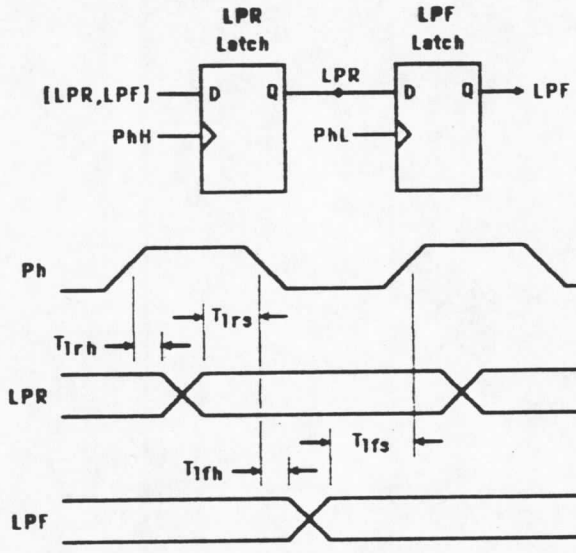
Signals external to chips are usually synchronized by a single-phase clock. In the Pixel-planes Engine, these clocks are **Ph** for image-generation signals and **Px** for video scan-out control, and **V** for video data. The signals **PhH** and **PhL** refer to the clock and its complement. Qualified clocks are written by joining the qualifying signal name with the clock name. A latch enable signal might be written **LEnHPhH**, for example.

L

Latched on rising or falling clock edge. **L**-type signals are fully named by joining the signal name with a four-character type. The four fields of the type are summarized in the following table.

Field	Allowed Char's	Function
1	**H,L**	Indicates active-HI or active-LO
2	**L**	Timing characteristic–see below
3	**P,X,V**	**Ph, Px,** or **V** clock
4	**R,F**	Rising or falling clock edge

Waveforms and timing are shown in Figure 2.1.9.



Timing Constraints for L-type Signals	
T_{1rh}	LPR signal must satisfy the hold-time requirements of an LPR latch
T_{1rs}	LPR signal must satisfy the setup-time requirements of an LPF latch
T_{1fh}	LPF signal must satisfy the hold-time requirements of an LPF latch
T_{1fs}	LPF signal must satisfy the setup-time requirements of an LPR latch

Figure 2.1.9: Timing constraints for L-type signals.

The timing constraints shown in Figure 2.1.9 refer to This notation refers to latches whose edge-triggered clock inputs are driven by the **Ph** clock and its inverse, respectively. The timing constraints state that an **LPR** signal must remain stable after the active clock edge for sufficient time to satisfy the hold-time requirements of an LPR latch to whose D input the typed signal is applied. Further, the signal must have stabilized before the opposite clock edge for sufficient time

to satisfy the setup-time requirements of a LPF latch to whose D input the typed signal is applied. This second requirement automatically guarantees that an LPR signal meets the setup-time requirement for an LPR latch. Signals input to chips are all of type **LPR**; detailed timing requirements for the chip input interface are given in section 2.1.3.2.

S

Stable on an edge. This signal type is defined to take care of chip outputs, which, since they emerge from a 2-phase environment, cannot easily be forced to meet the requirements for either **LR** or **LF**. The four-character type field is identical to **L**-type with **L** replaced by **S**. Only **SR** signals, however, are actually allowed to emerge from chips. Timing constraints for **SPR** are similar to **LPR**, but weaker: the hold time for **SPR** is the same as **LPR**, while the setup time is the same as **LPF**.

T

This type applies to tri-state inputs/outputs. The full four-character type is identical to that for **L**-type; if a clock edge is specified in the last type field, the signal may change state to HI, LO, or HI-Z at that edge.

2.1.1.4. Module Interconnection Rules

The following set of rules attempts to guarantee the correct construction and interconnection of design modules. In order to provide a simple scheme for specifying the legal interconnections of modules, the following rule is invoked:

Rule 1: Modules may only be composed of modules or of circuit primitives, not both.

This implies that the lowest level module (leaf cell) will contain nothing but circuit primitives, and all higher-level modules (composition cells) will contain nothing but sub-modules (leaf cells and composition cells, no primitives). Design module inputs and outputs are governed by:

Rule 2: Inputs and outputs of modules may only of the primary types Clock, S, and E.

This rule allows only nodes *within* a leaf cell to have other (ancillary) types. Module interconnections are governed by

Rule 3: An output of a given type may only be connected to inputs of the same type.

It is not always convenient to translate the circuit/logic representation of a design module directly into a single module in chip geometry. The following rule is invoked:

Rule 4: The boundary of a circuit-level design module must correspond exactly to the boundary of a leaf or composition cell at the geometric level. Rule 2 applies to geometric design modules.

In chips where two asynchronous clocks are used, the following rule applies:

Connections between Pn and Xn signal types are strictly forbidden.

Modules which have pins timed on the two different clocks should be designed *very* carefully!

2.1.2. Schematic Symbol Conventions

Schematics in this design will use conventional logic symbols wherever possible. A few symbols used in gate- and transistor-level descriptions for CMOS circuits are not well standardized; the conventions used in this document are shown in Figure 2.1.10.

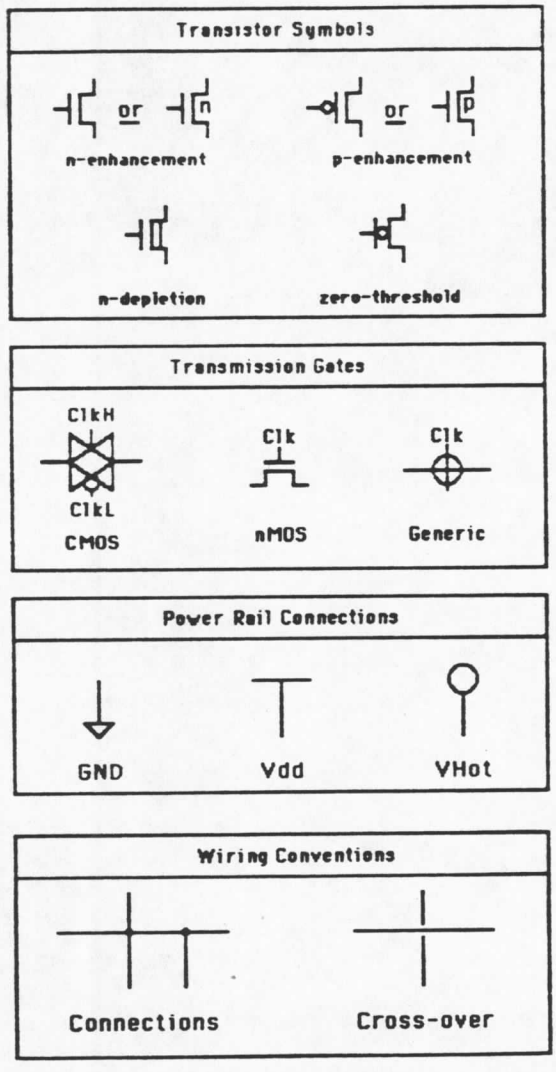


Figure 2.1.10: Schematic Symbol Conventions.

2.1.3. Chip External Interface Design

2.1.3.1. Clocking Strategy

Outside the Pixel-planes enhanced memory chip, the image-generation processes in the Pixel-planes Graphic Engine are timed by the **Ph** (single-phase) clock, and all pixel scan-out operations are timed by the **Px** clock. All control inputs to the chip are of type **LPR** or **LXR**, with special restrictions described below in section 2.1.3.2. Video data outputs from the Pixel-planes enhanced memory chip are of type **EX2**; other outputs are of type **SPR** or **SXR**. The details of interfacing these signals to external circuitry are described in 2.1.3.3.

This section describes the relationship between off-chip single-phase clocks and the on-chip 2-phase clocks. It also outlines how on-chip qualified clocks are generated.

On-Chip 2-Phase Clock Generation

The conversion from single-phase bus clock to internal two-phase non-overlapping clock is performed as suggested by Chuck Seitz in Chapter 7 of Mead and Conway's "Introduction to VLSI Systems." Figure 2.1.11 shows the logic and timing for this circuit. The clock generator module inside the chip generates **Ph1** and **Ph2**, accepting as inputs one or more qualified clocks

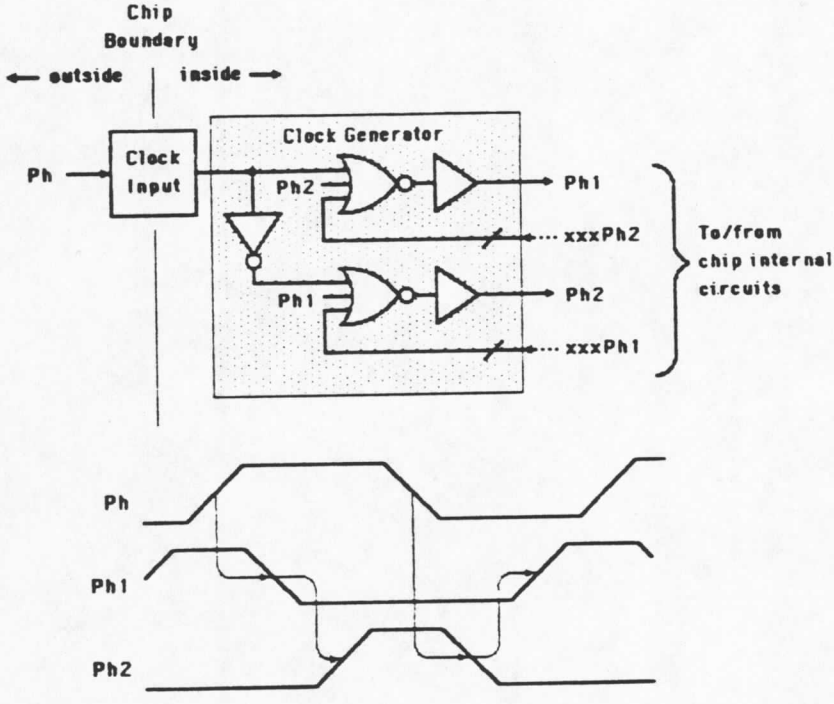


Figure 2.1.11: Converting single-phase clock to 2-phase non-overlapping clock; arrows on timing diagram show causality.

on each phase, denoted **xxxPh1** and **xxxPh2** in Figure 2.1.11. The generator is basically a cross-coupled circuit that prevents **Ph1** from being asserted until **Ph2** has returned LO, and *vice-versa*. It may be possible to select a 'worst-case' qualified clock, whose path contains the maximum delay across the entire chip, for each phase, thereby guaranteeing non-overlap for all clocks and qualified clocks.

On-Chip Clock Distribution and Qualification

The Pixel-planes design does not use the usual pyramid-style clock distribution scheme for generating and re-amplifying clock signals. Rather, the clock buffers in the clock generator of Figure 2.1.11 are sized large enough to drive the entire clock load on the chip. Clock signals are treated next in importance to the supply rails for on-chip wiring. They are distributed in uninterrupted metal so far as possible, with polysilicon or low-resistance-wire crossovers only at power rails. In CMOS designs with second metal, it should be possible to distribute clocks entirely in metal. All unqualified clock inputs to modules within the chip are driven directly from this distribution system, with only short polysilicon links allowed.

Qualified clocks are used extensively in this design for control of registers, selectors, and so forth. All qualified clocks are generated in a way that introduces minimum propagation delay and minimum skew with respect to the parent clock. This method involves inserting a pass-gate structure between the parent clock and the qualified output. This pass-gate is sized large enough to introduce acceptably small skew between the parent and qualified clock, and its control gate(s) may be bootstrapped to improved performance.

Figure 2.1.12 shows a CMOS design for a qualified clock generator that provides two complementary signals. For roughly the same transistor count and circuit area, this transmission-gate circuit has a considerably smaller propagation time than a functionally equivalent complementary AND circuit. The expected waveforms for the clock and qualified clock

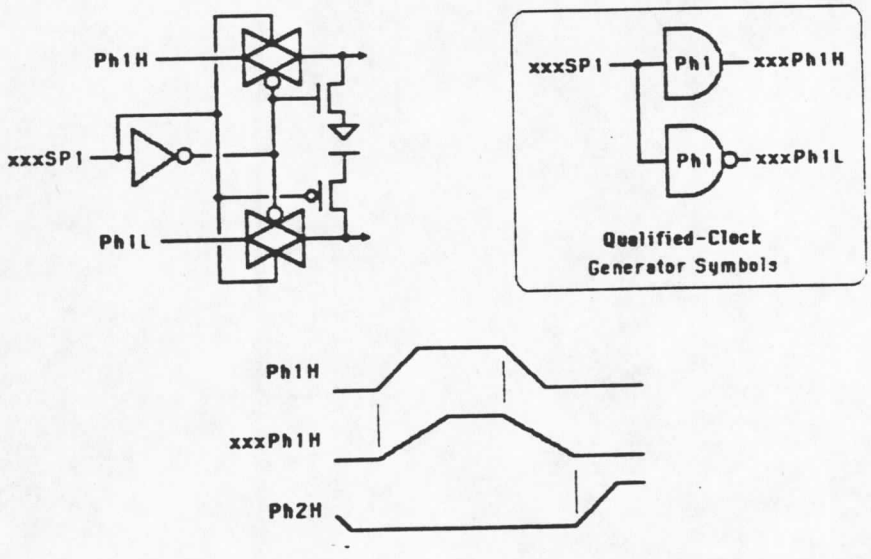


Figure 2.1.12: CMOS qualified clock generator (no bootstrapping).

are shown in the Figure; the delay through the circuit is dominated by the slope of the rising and trailing edges. These slopes are determined (roughly) by the product of the ON channel resistance of the transistors in the transmission-gate and the load capacitance, and they can be easily controlled (and matched to other qualified clock generators) by appropriately sizing the two transistors in the transmission gate.

The bootstrapped equivalent of this scheme is based on an approach suggested by Chuck Seitz and used extensively in the CalTech MOSAIC processor. A boot-strapped qualified clock generator circuit for nMOS is shown in Figure 2.1.13.

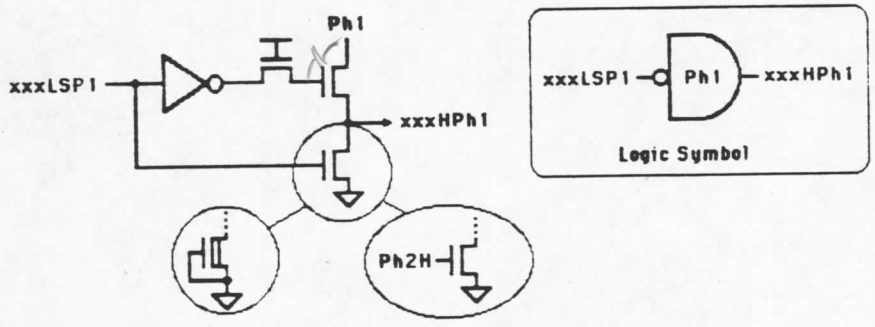


Figure 2.1.13: Boot-strapped qualified clock driver.

In the circuit of Figure 2.1.13, the qualified clock output should be "snubbed" to ground when not asserted, since the node is otherwise floating. Three methods for accomplishing this function are shown.

The performance of bootstrapped clock qualifiers is greatly increased by using so-called "hot clocks". Hot clocks, toggling between GND and VHot (typically 2-3 volts above Vdd) are generated on-chip and distributed throughout the chip. Several advantages are obtained from hot-clocking: nMOS pass-gates, single n-enhancement transistors, are enabled to pass a logic-HI without a threshold drop and with considerably smaller delay. Without the threshold drop usually associate with pass-gate logic, level-translating inverters are not needed. In nMOS circuits, k=4 inverters can be used to amplify steered nodes, with attendant reduction of area and increase in speed.

Boot-strapped clock qualifiers must be laid out with some care in order to achieve good performance. The bootstrapping capacitance is essentially the overlap capacitance between the drain and gate of the clock-pass transistor. The bootstrapping efficiency is a function of the ratio of this capacitance to the parasitic capacitance attached to the bootstrapped gate. The small isolation transistor, and the connection between it and the bootstrapped node (which necessarily involves a change from the diffusion to the polysilicon layer) must be made as compact as possible. The bootstrap transistor must be "large enough" to obtain a reasonable bootstrapping efficiency; we have found a 1/8-square device to be the smallest reasonable size. A metal 'cap' can be connected to the drain and extended over the gate to increase the bootstrapping capacitance.

2.1.3.2. Chip I/O Interface for Static Signals

This section defines the timing and circuit requirements for passing 'static' (as opposed to precharge-evaluate signals, which are circuitry.

Figure 2.1.14 shows details of timing for an input interface that brings a type **LPR** signal onto the chip, where it must become a type **SP1**.

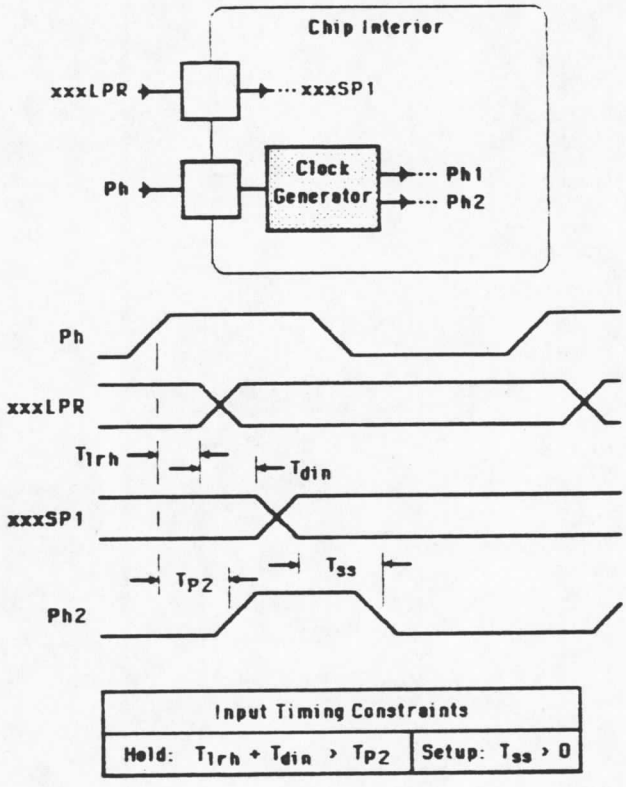


Figure 2.1.14: Details of chip input interface.

A two-sided timing constraint applies to inputs to the chip. The 'hold' constraint says that the **LPR** input must be delayed sufficiently beyond the rising clock edge so that the internal **SP1** signal does not change until after the leading edge of **Ph2**. This constraint will likely require that the latch that produces the **LPR** signal be 'slow' in propagating signals to its data outputs. The other side of the timing constraint simply requires that the total delay between the rising **Ph** edge and settling the **SP1** signal will be short enough to satisfy the setup time requirements for the **SP1** type.

Figure 2.1.15 shows timing for an output interface that brings a type **SP1** signal off-chip, where it becomes a type **SPR**. A single-side constraint is sufficient to guarantee correct timing (the hold-time requirement for the **SPR** output shown is causally guaranteed by the timing characteristics of the **SP1** signal). The constraint, shown on Figure 2.1.15, simply states that the the **SPR** signal must be stable sufficiently far ahead of the leading edge of **Ph** to satisfy the setup-time requirements of the latch which is to capture the signal. The constraint is specified in terms of the delay between falling edge of **Ph** and falling edge of **Ph2**, the setup time for **SP1** signals, and the delay through the output pad. This specification focuses on the circuit elements

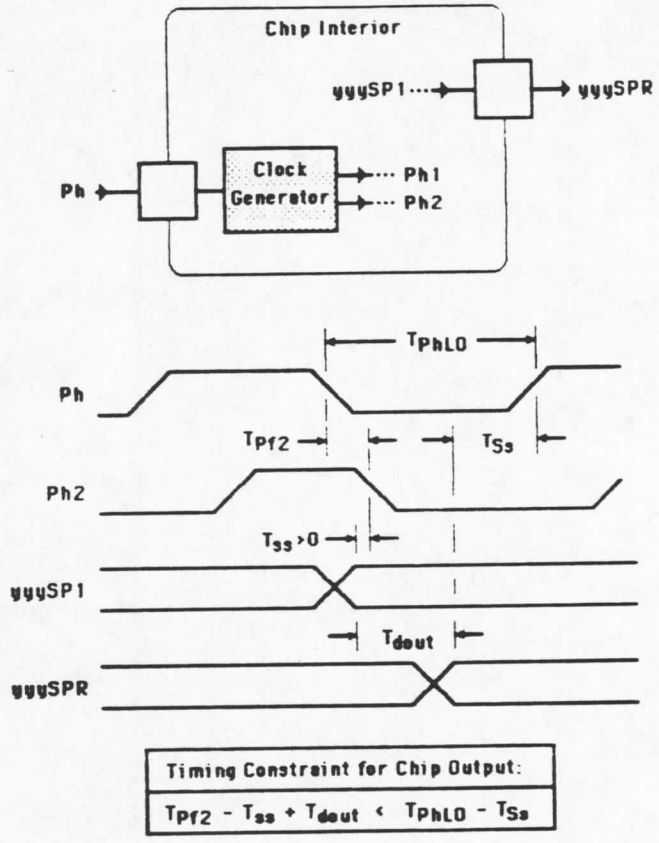


Figure 2.1.15: Timing constraints for chip output interface (static).

that must be modelled in order to satisfy the constraint.

2.1.3.3. Chip I/O Interface for Dynamic Signals

This section describes interface requirements for precharge/evaluate signals that pass through the chip external interface circuitry. (The video data outputs of the enhanced memory chips are of this type.)

Figure 2.1.16 shows circuit details and timing for capturing an **E**-type chip output at an external edge-triggered latch. The timing constraint states that the evaluation phase for the **E**-type signal shall be complete sufficiently ahead of the rising **Px** edge to satisfy the setup-time requirements of the latch which captures the signal. The external interface for **E**-type signals should use a latch from the 74HC logic family, since these devices feature very high input impedance (so as not to discharge the dynamic node) and logic threshold of 2.5 volts. A pull-up resistor of a few thousand ohms may be needed to damp out ringing on the wire carrying the **E**-type signal.

E-type signals may be input to chip pins as well. Since these signals can only originate on another chip operating on the same clock, the interface design and timing closely resembles

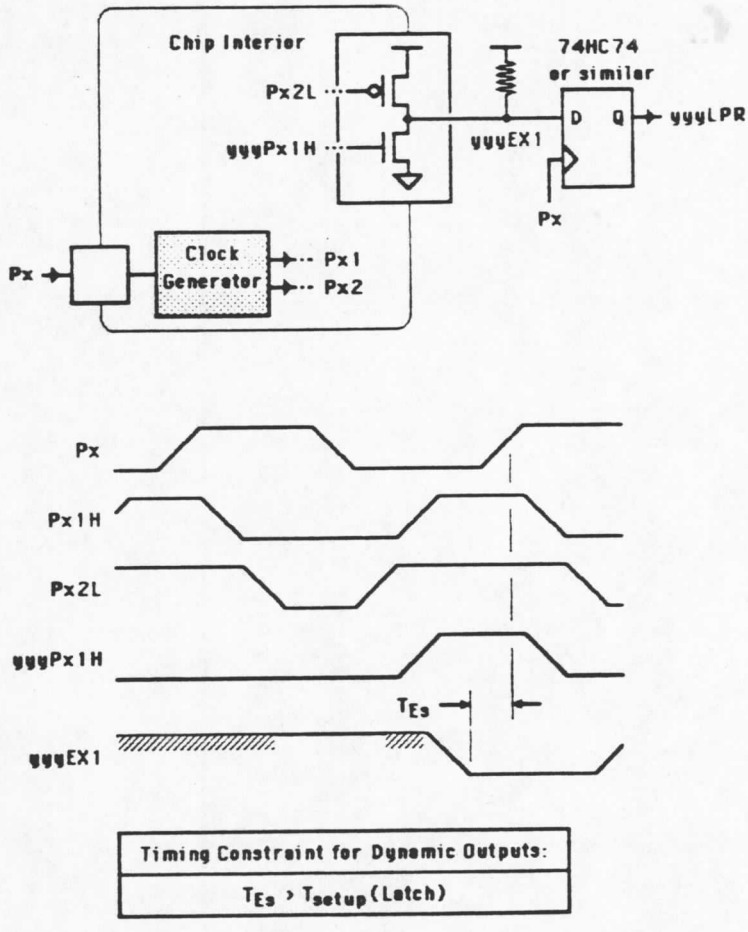


Figure 2.1.16: Circuit details and timing for external interface for dynamic outputs.

interfaces between on-chip modules that communicate using E-type. Refer to section 2.1.1.2 for details.