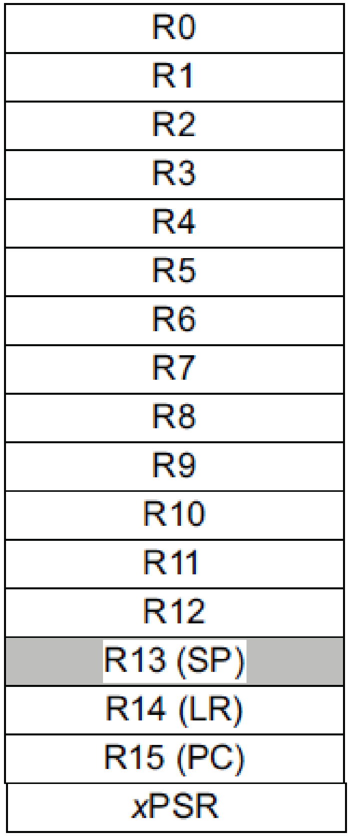
EECS 373

Introduction to Embedded System Design

Robert Dick
University of Michigan

Lecture 5: ABI, APB, and Build Process

30 Jan and 1 Feb 2024



R0
R1
R2
R3
R4
R5
R6
R7
R8
R9
R10
R11
R12
R13 (SP)
R14 (LR)
R15 (PC)
*x*PSR

Review based on questions in office hours / lab

- Bit-level manipulation
- Function calls and argument passing

Outline

- **Application Binary Interface**
- Advanced Peripheral Bus
- Deferred Details on Build Process

ABI summary

Detailed version

- Pass: r0-r3
- Return: r0 or r0-r1
- Callee saved variables: r4-r8, r11, maybe r9, r10
- Static base: r9 (might offset from this to write)
- Stack limit checking: r10 (SP >= r10)
- Veneers, scratch: r12 (lillypad)
- Stack pointer: r13
- Link register (function call return address): r14
- Program counter: r15

Simple version

- Callee preserves r4-r11 and r13
- Caller preserves r0-r3

ABI example

- main() calls f().
- f() calls g().
- g() calls h().

- What if f() clobbers r2?
- f() clobbers r5?
- g() clobbers r0?
- g() clobbers r4?
- h() clobbers r1?
- h() clobbers r6?

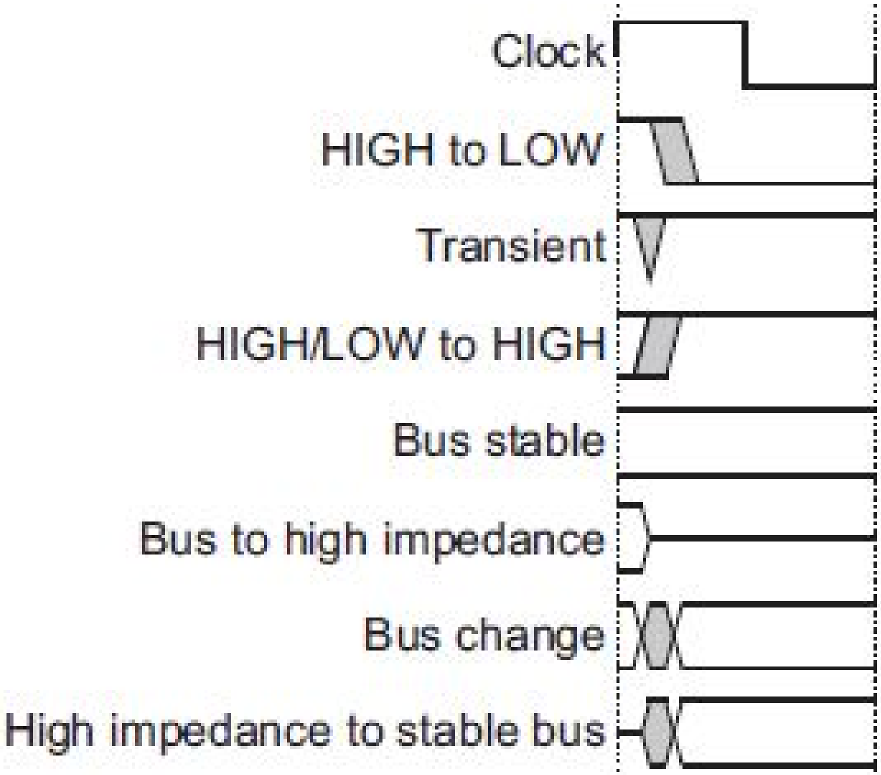
Outline

- Application Binary Interface
- **Advanced Peripheral Bus**
- Build Process

APB is designed for ease of use

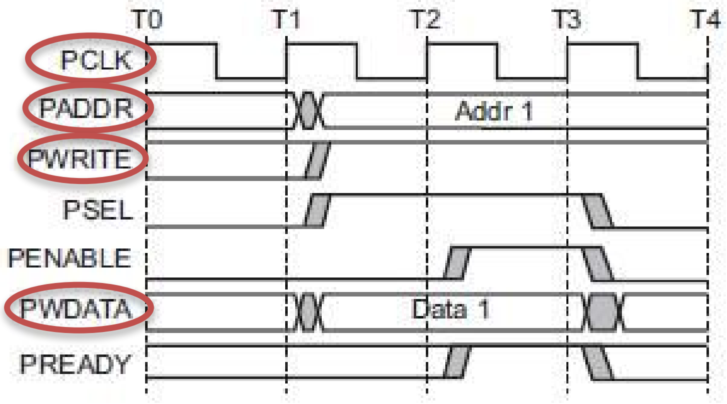
- Low-cost.
- Low-power.
- Low-complexity.
- Low-bandwidth.
- Non-pipelined.
- Ideal for peripherals.

APB Writes: Notation



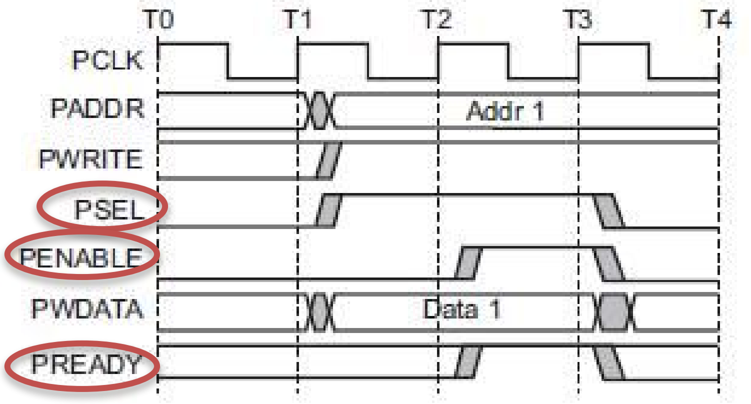
APB bus signals

- PCLK
  - Clock.
- PADDR
  - Address on bus.
- PWRITE
  - 1=Write, 0=Read.
- PWDATA
  - Data from processor.
- PRDATA
  - Data to processor.



APB bus signals

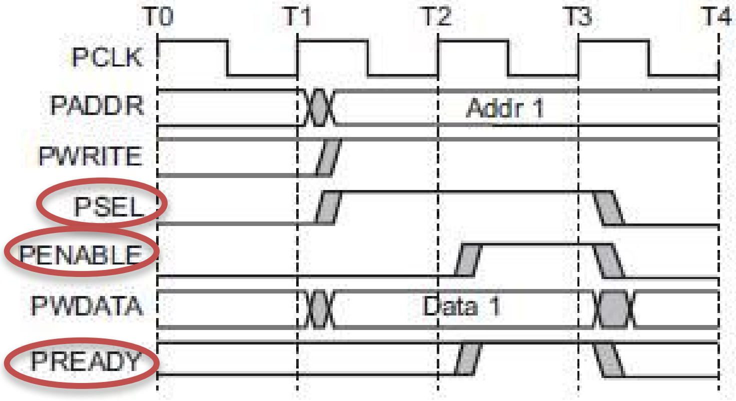
- PSEL
  - Asserted if the current bus transaction is targeted to ***this*** device.
- PENABLE
  - High during entire transaction *other than* the first cycle. Distinguishes between idle, setup, and ready.
- PREADY
  - Driven by target. Similar to #ACK. Means target is *ready*.
  - Each target has it's own PREADY line.



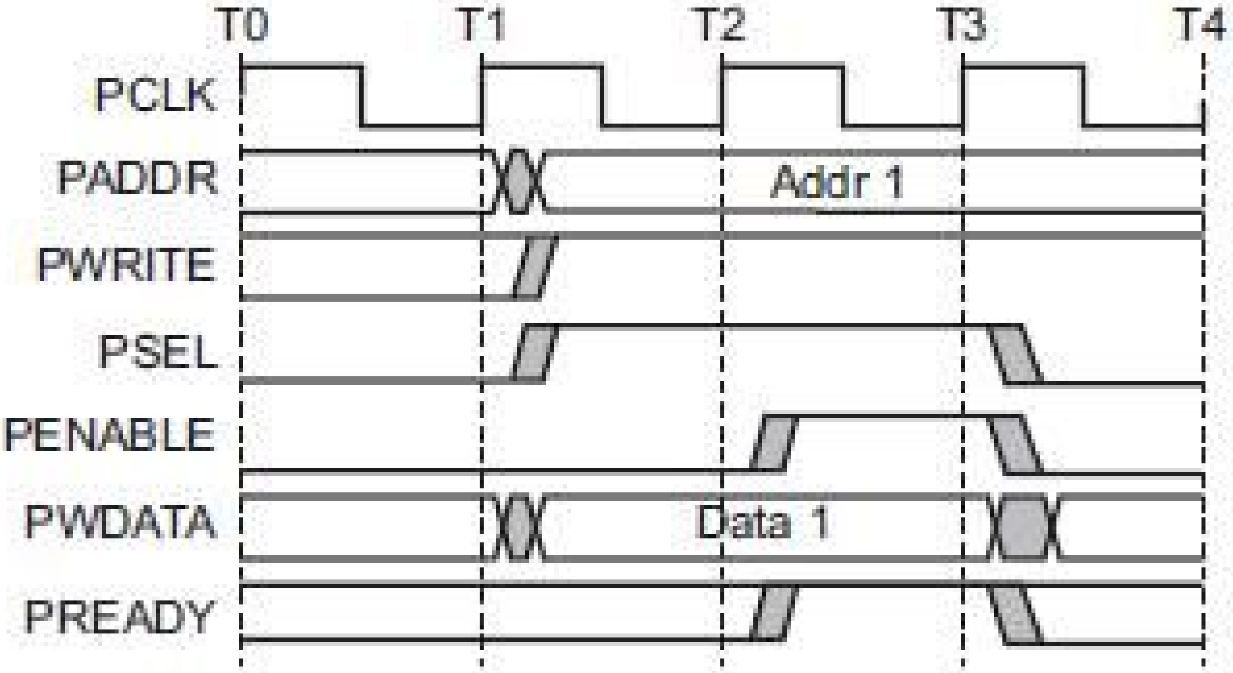
Sharing

- Unshared.
  - PSEL.
  - PREADY.
  - PRDATA.
  - PSLVERR.

- Shared: everything else
  - PCLK.
  - PADDR.
  - PWRITE.
  - PENABLE.
  - PWDATA.



What is happening?

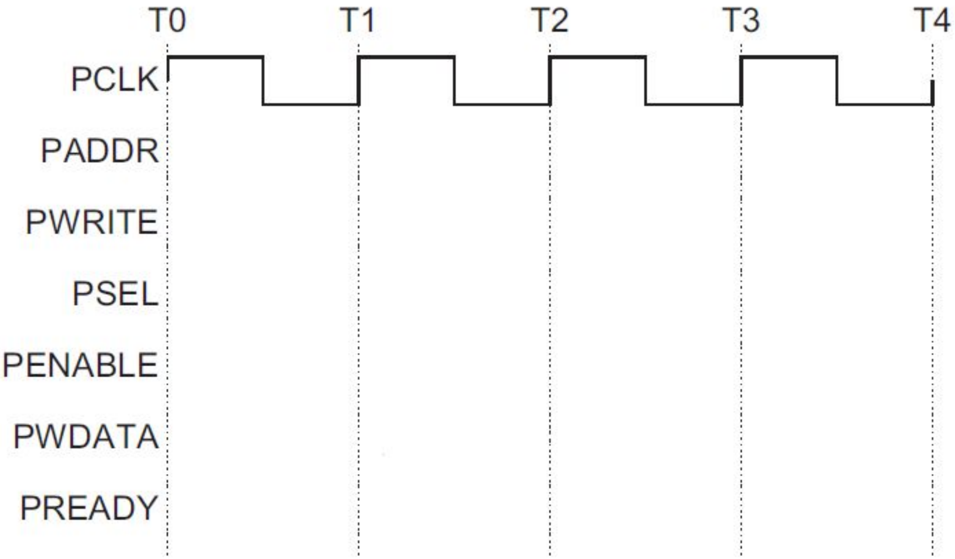


Example setup

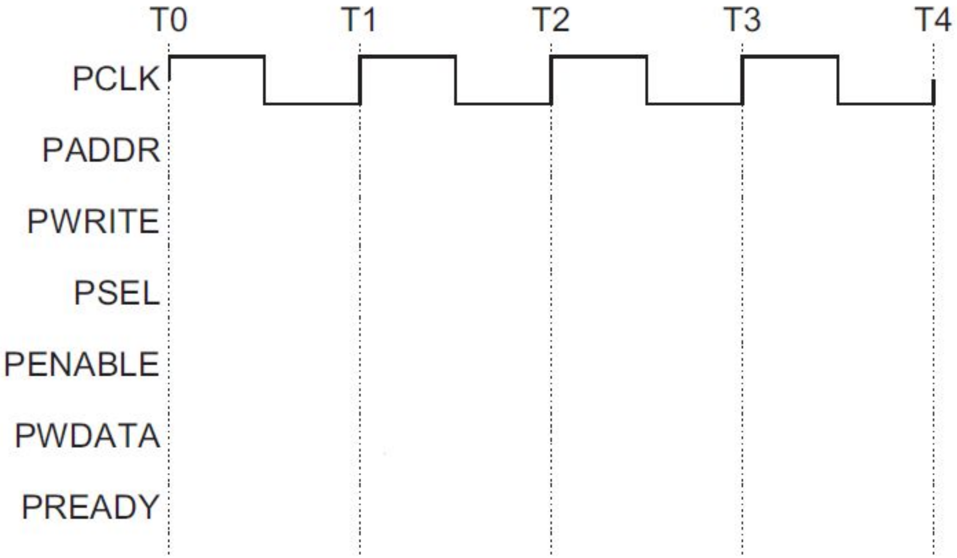
- Assume one initiator “CPU” and two target finite state machines (D1 and D2)
- D1 is mapped to address 0x00001000-0x0000100F
- D2 is mapped to 0x00001010-0x0000101F

CPU stores to 0x00001004 w.o. stalls

D1



D2



LSB of register controls LED

PWDATA[31:0]

PWRITE

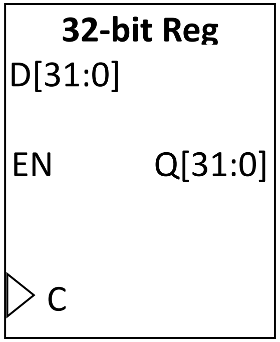
PENABLE

PSEL

PADDR[7:0]

PCLK

PREADY



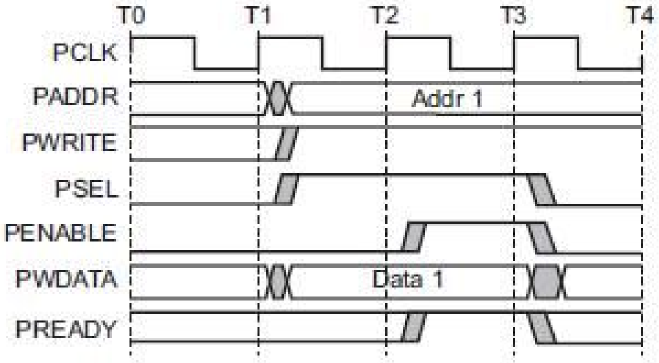
32-bit Reg

D[31:0]

EN

Q[31:0]

C



Assuming APB only gets lowest 8 bits of address

Reg A should be written at address 0x00001000
Reg B should be written at address 0x00001004

PWDATA[31:0]

PWRITE

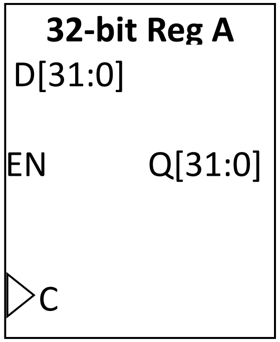
PENABLE

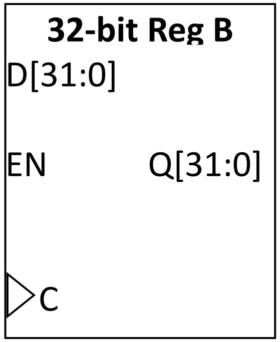
PSEL

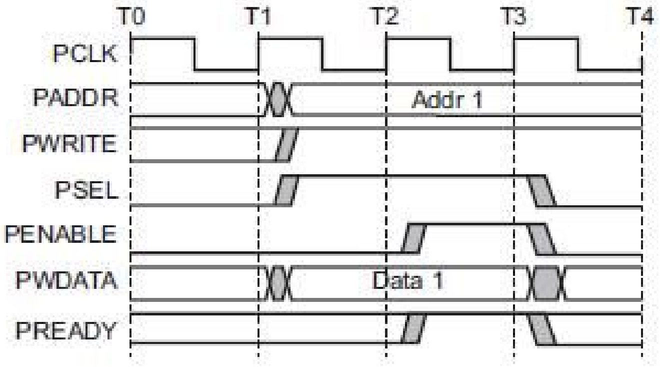
PADDR[7:0]

PCLK

PREADY

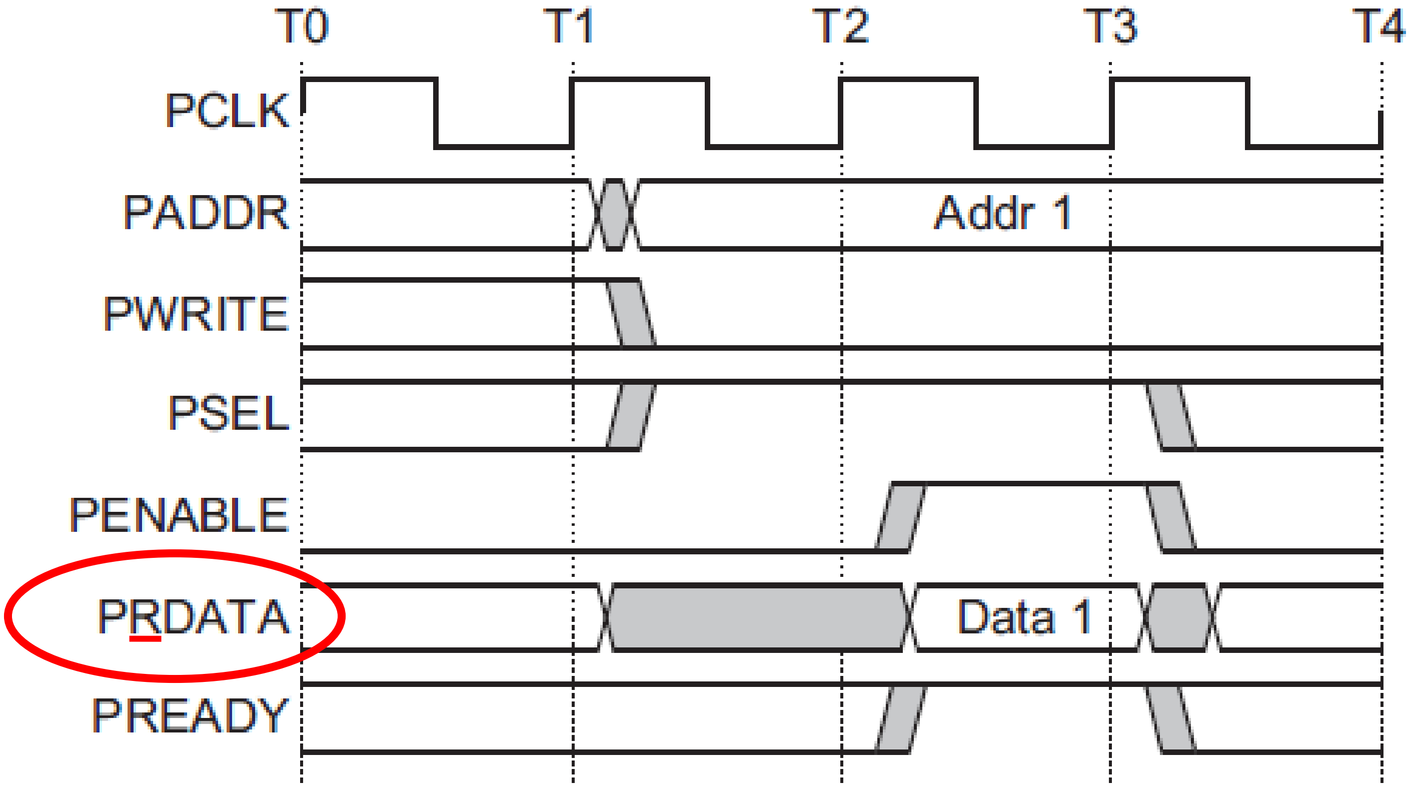






Assuming APB only gets lowest 8 bits of address

Reads



Each follower device has its own read data (PRDATA) bus.

Recall that “R” is from the initiator’s viewpoint—the device drives data when read.

Device provides data from switch for any of its addresses

PRDATA[31:0]

PWRITE

PENABLE

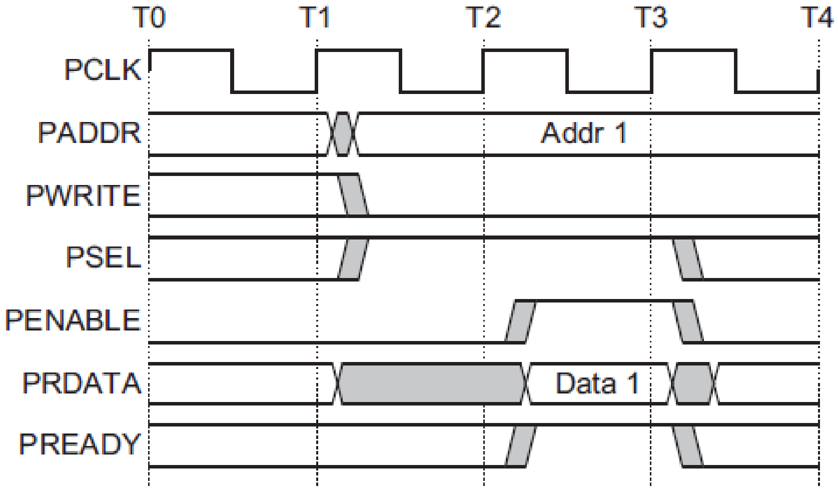
PSEL

PADDR[7:0]

PCLK

PREADY

Switch



Switch A for 0x00001000, B for 0x00001004

PRDATA[31:0]

PWRITE

PENABLE

PSEL

PADDR[7:0]

PCLK

PREADY

Switch A

Switch B

All reads read from register, all writes write

PWDATA[31:0]

PWRITE

PENABLE

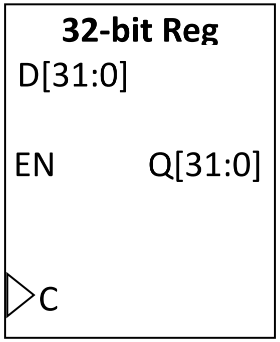
PSEL

PADDR[7:0]

PCLK

PREADY

PRDATA[31:0]



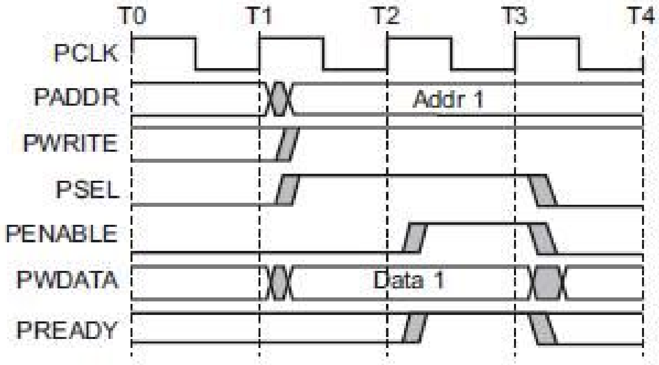
32-bit Reg

D[31:0]

EN

Q[31:0]

C



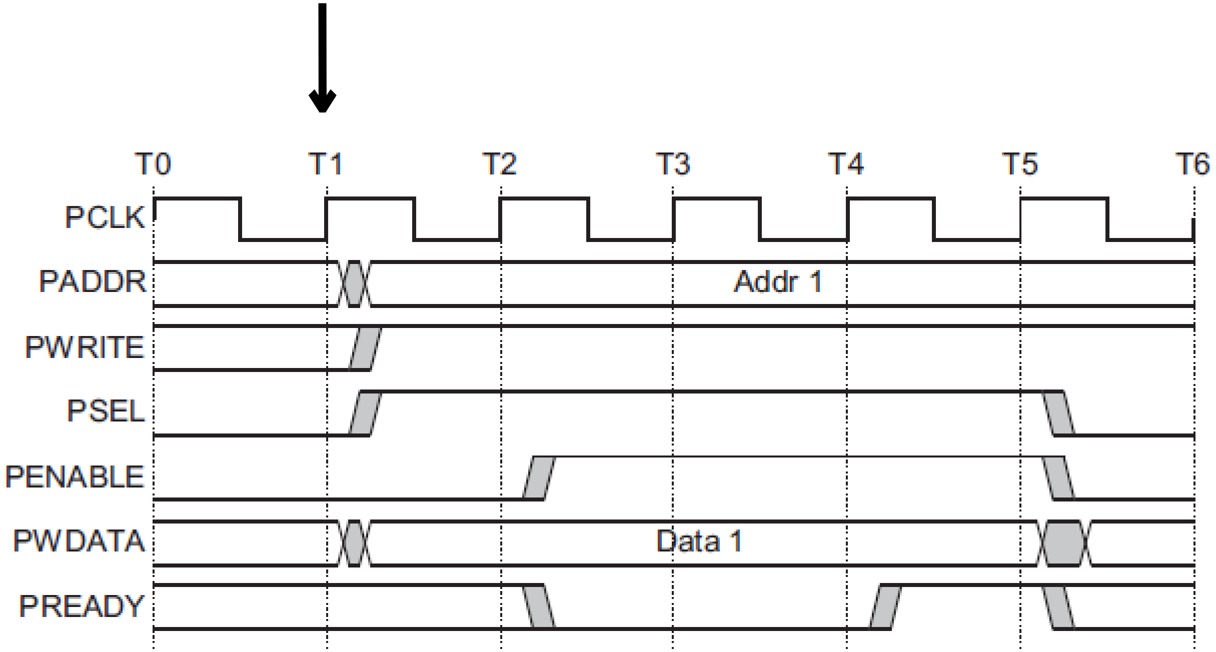
Assuming APB only gets lowest 8 bits of address

Errors and stalling

- PSLVERR high on error.
  - Otherwise, ground it.
- Use PREADY to stall.
  - > a few cycles probably implies design flaw.

A write transfer with wait states

Setup phase begins
with this rising edge



Setup Phase

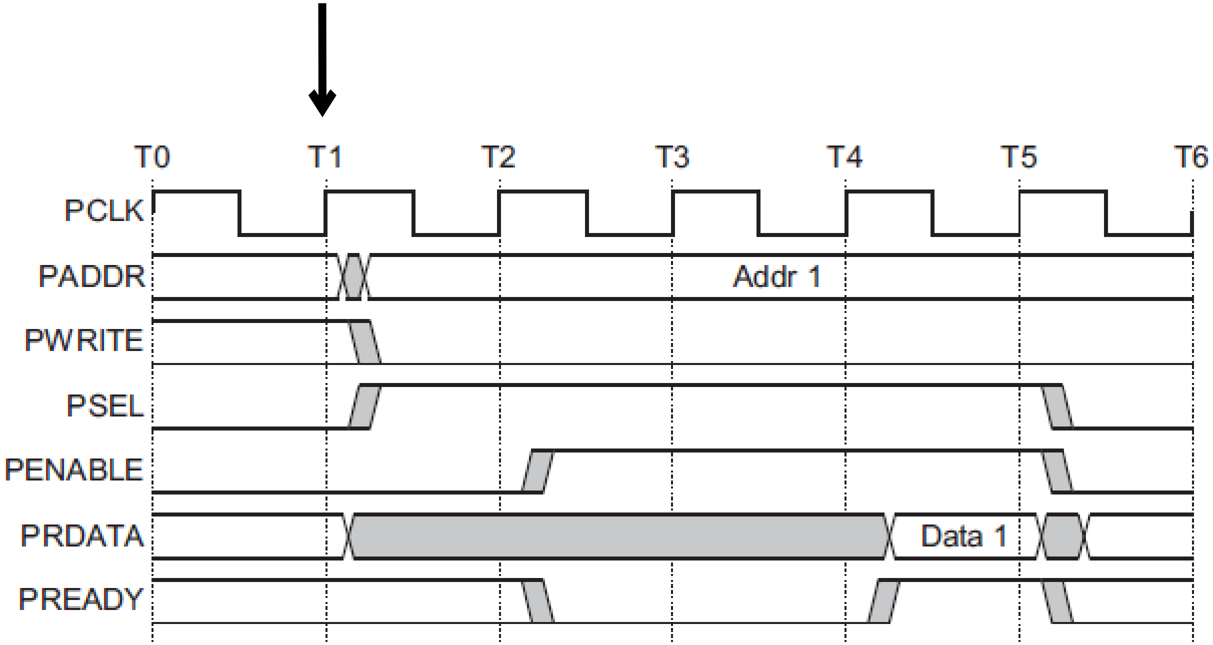
Wait State

Wait State

Access Phase

A read transfer with wait states

Setup phase begins
with this rising edge



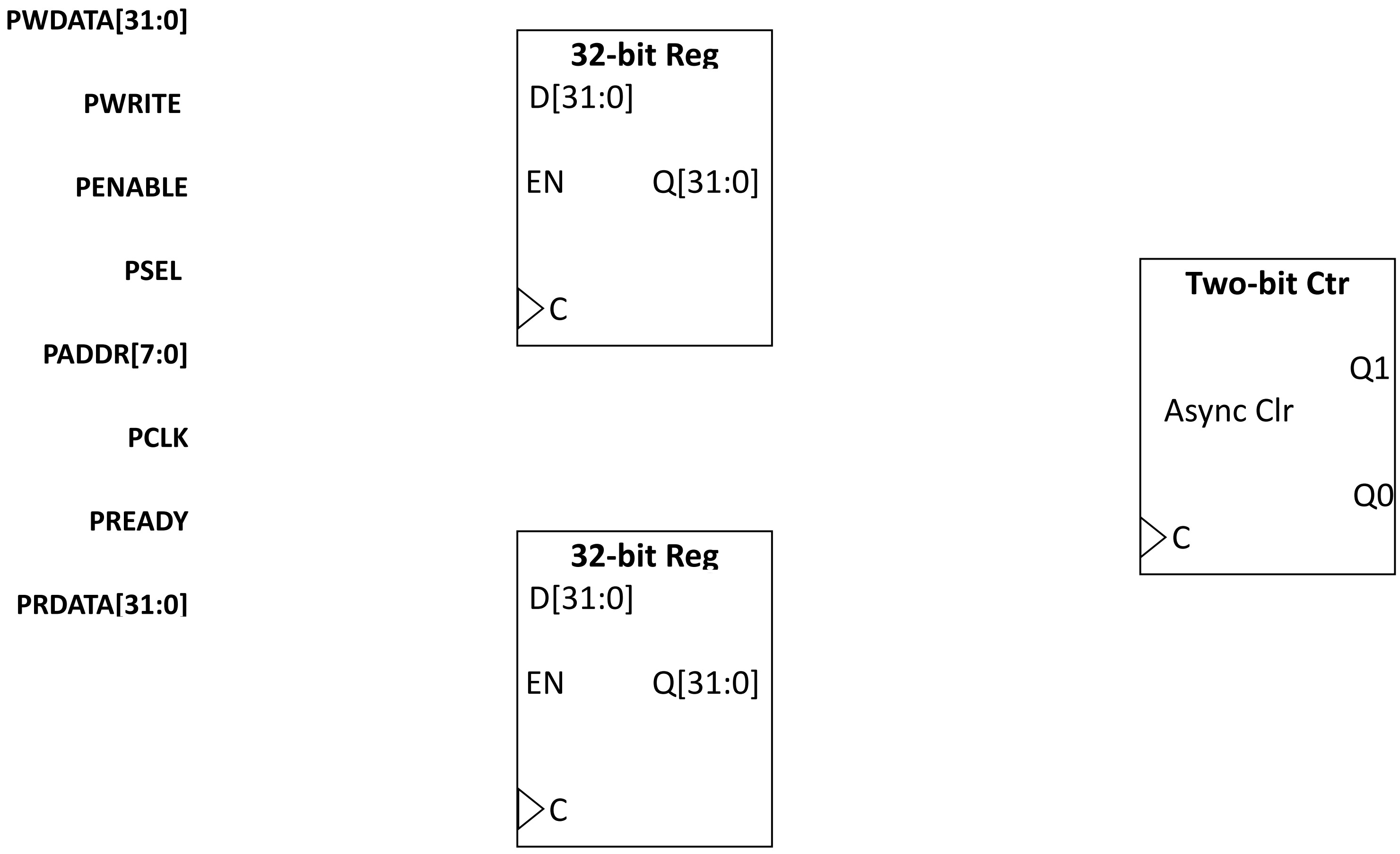
Setup Phase

Wait State

Wait State

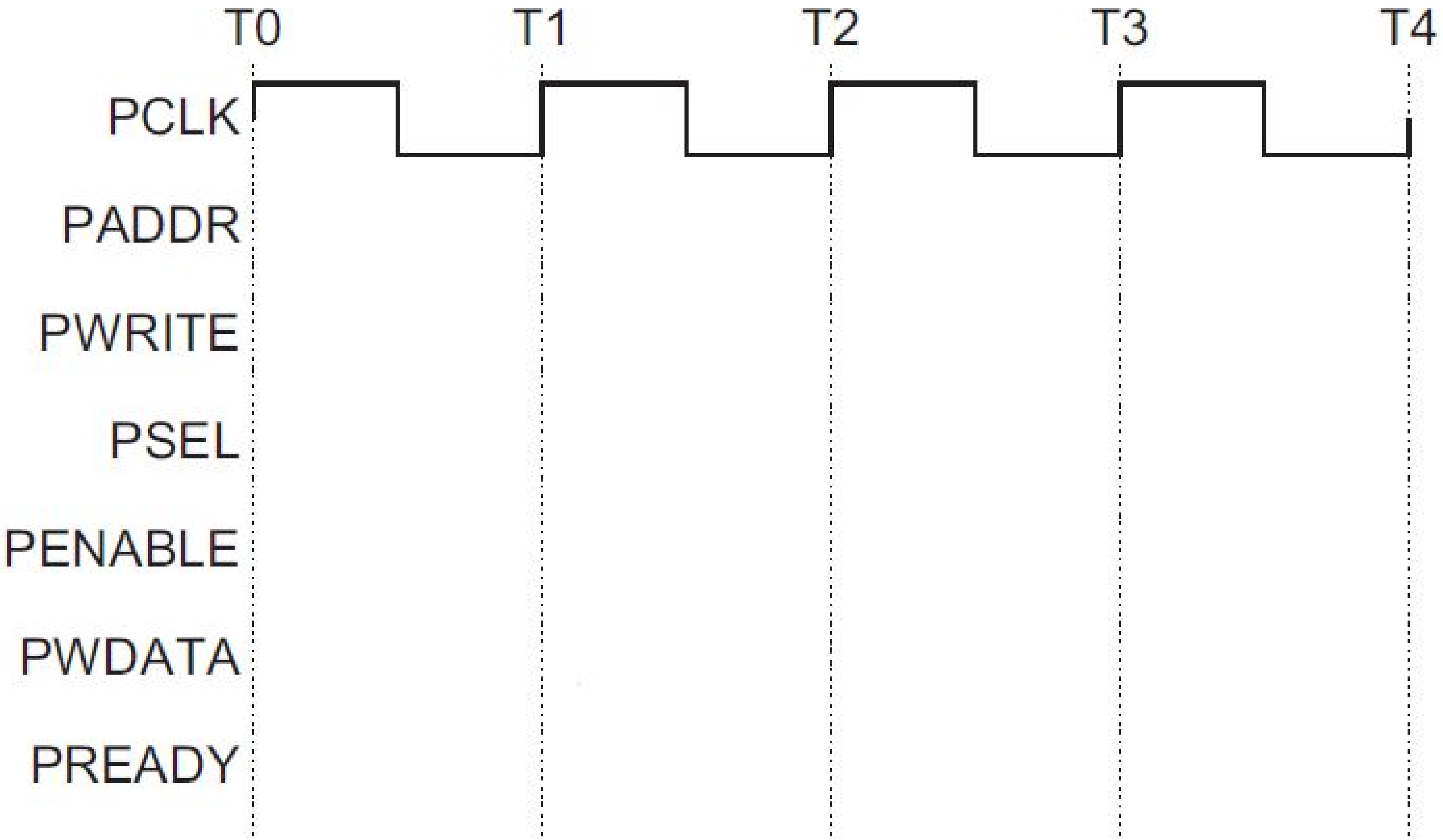
Access Phase

Writes to 0x00000002 go to fast local servo, writes to 0x00000004 go to slow remote servo



Assuming APB only gets lowest 8 bits of address

Timing diagram for 0x00000004 write



Additional capabilities

- There is another signal, PSLVERR which we can drive high on failure.
- Tie that to 0 if failure impossible.
- Assuming that our device never stalls.
  - We **could** stall if we needed.
  - PREADY.

Verilog

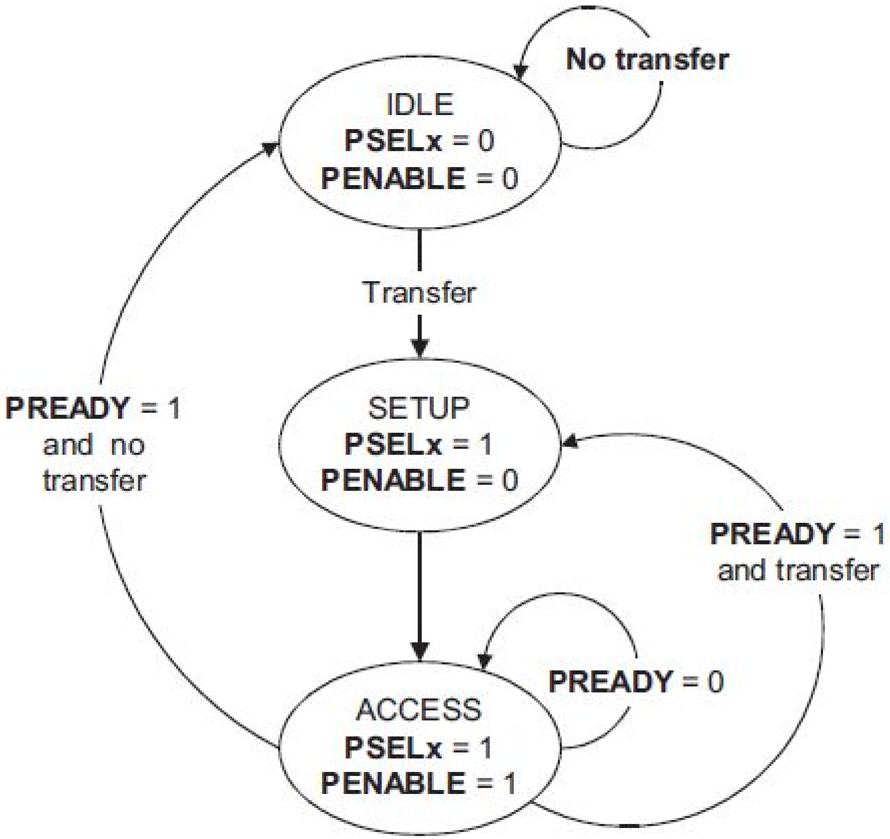
```
/*** APB3 BUS INTERFACE ***/
input PCLK,                   // clock
input PRESERN,                // system reset
input PSEL,                   // peripheral select
input PENABLE,                // distinguishes access phase
output wire PREADY,           // peripheral ready signal
output wire PSLVERR,          // error signal
input PWRITE,                 // distinguishes read and write cycles
input [31:0] PADDR,           // I/O address
input wire [31:0] PWDATA,     // data from processor to I/O device (32 bits)
output reg [31:0] PRDATA,     // data to processor from I/O device (32-bits)

/*** I/O PORTS DECLARATION ***/
output reg LEDOUT,            // port to LED
input SW                      // port to switch
);

assign PSLVERR = 0;
assign PREADY = 1;
```

APB state machine

- IDLE
  - Default APB state
- SETUP
  - When transfer required
  - PSELx is asserted
  - Only one cycle
- ACCESS
  - PENABLE is asserted
  - Addr, write, select, and write data remain stable
  - Stay if PREADY = L
  - Goto IDLE if PREADY = H and no more data
  - Goto SETUP is PREADY = H and more data pending



Outline

- Application Binary Interface
- Advanced Peripheral Bus
- **Build Process**

Pickmin

- What does a Makefile do?
- Linker scripts.
- Refresher on build process.
- Two paths to executable (C, assembly).

Stack-Call

- Detailed walk-through on stack allocation and deallocation.
- Double-word return.

Veneers

- Lillypad.
- PC contains current instruction + 8 bytes.

Weak references

- Allows you to conditionally call functions.
- May be useful in labs and projects.
- Understand linking better.

What does a weak symbol imply?

- Provides a default entry in a function vector.
- Why useful? Allows override at link time.

What does a call through a weak symbol imply?

- If the symbol exists, call function.
- If not, do nothing.
- Allows link-time conditional calls. No recompilation.
  - Large projects w. libraries and multiple build versions.

Weak symbols

- Recall function pointers.
- Can define symbol that will be clobbered by later definitions of same symbol.
- Useful for interrupt vectors or custom, e.g., cleanup code.

Done.