Testability

UNIVERSITEIT GENT

Dimitri Van Cauwelaert
Michaël Heyvaert

Testability

- specification => design ?

- Verification (internal process)

    Did we build it right?

    => testbenches, ...

- Validation (external process)

    Did we built the right thing

    => acceptance procedure with customer

Testability

- specification => design ?

I

- Verification (internal process)

  Did we build it right?

  => testbenches, ...

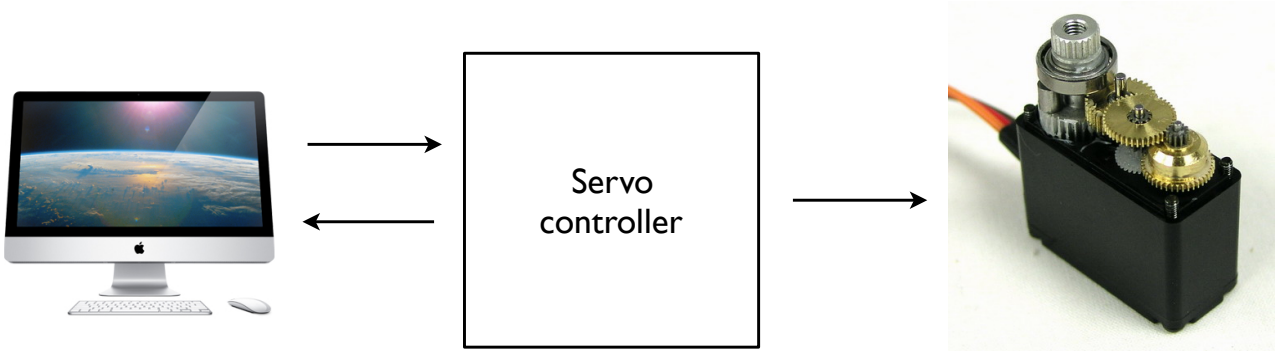
II

- Validation (external process)

  Did we built the right thing

  => acceptance procedure with customer

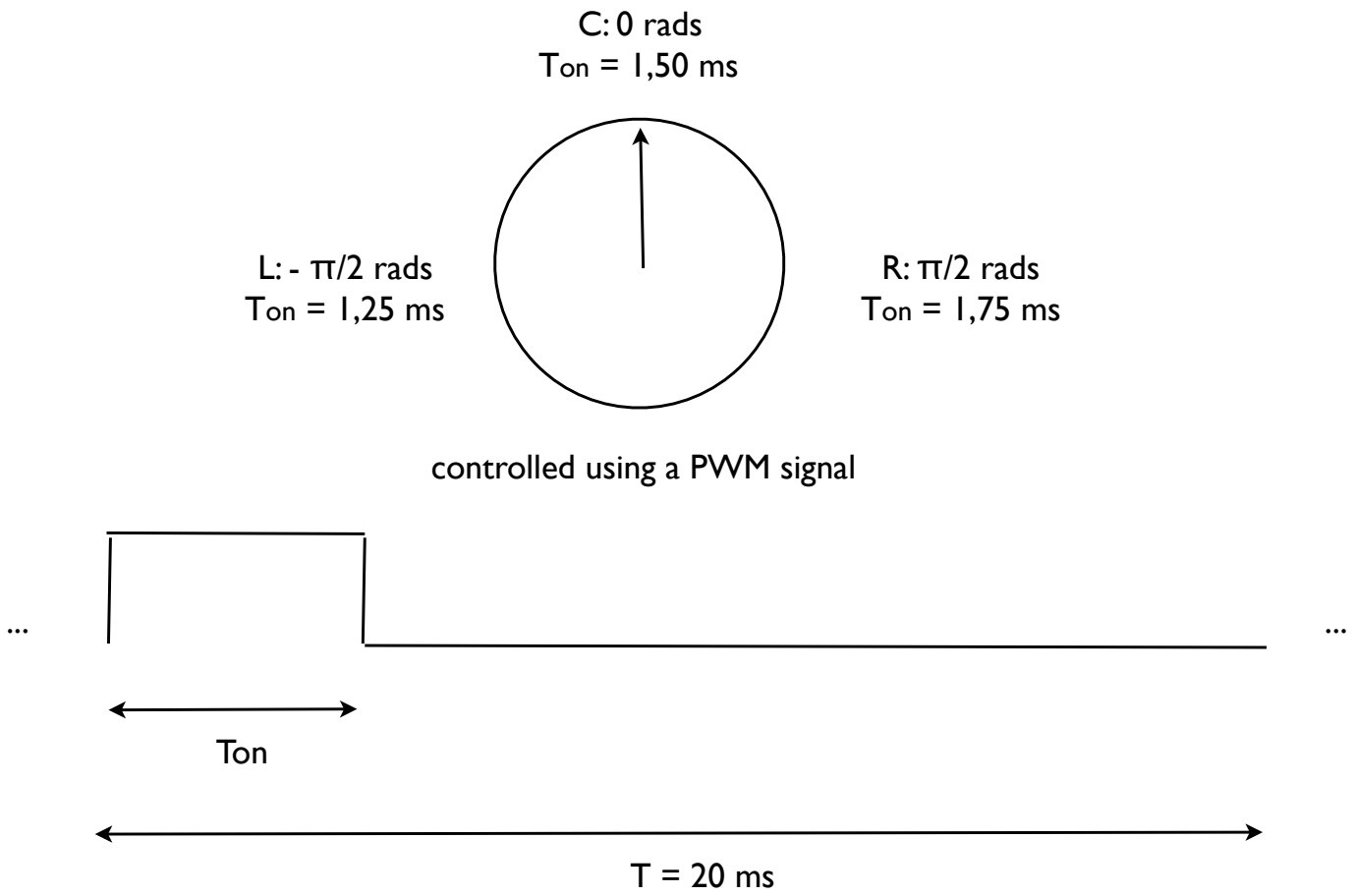
Design: servo controller



µC

Servo motor

Servo motor



Building the PWM signal

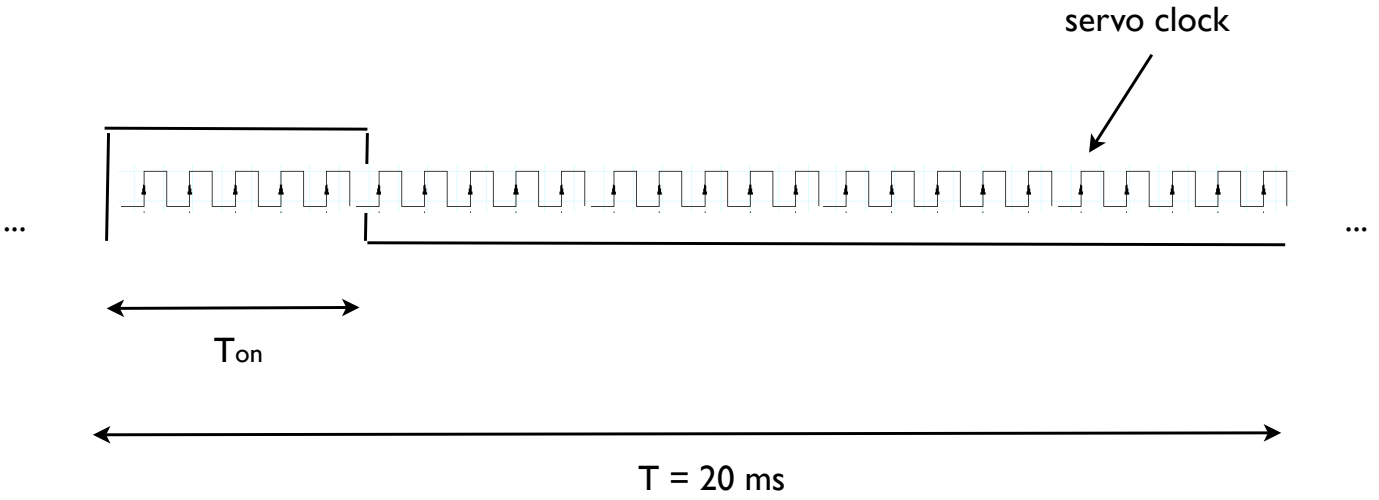
$T_{sc} << T_{clk}$

$T_{on} = (n_{offset} + n_{positie}) \times T_{sc}$

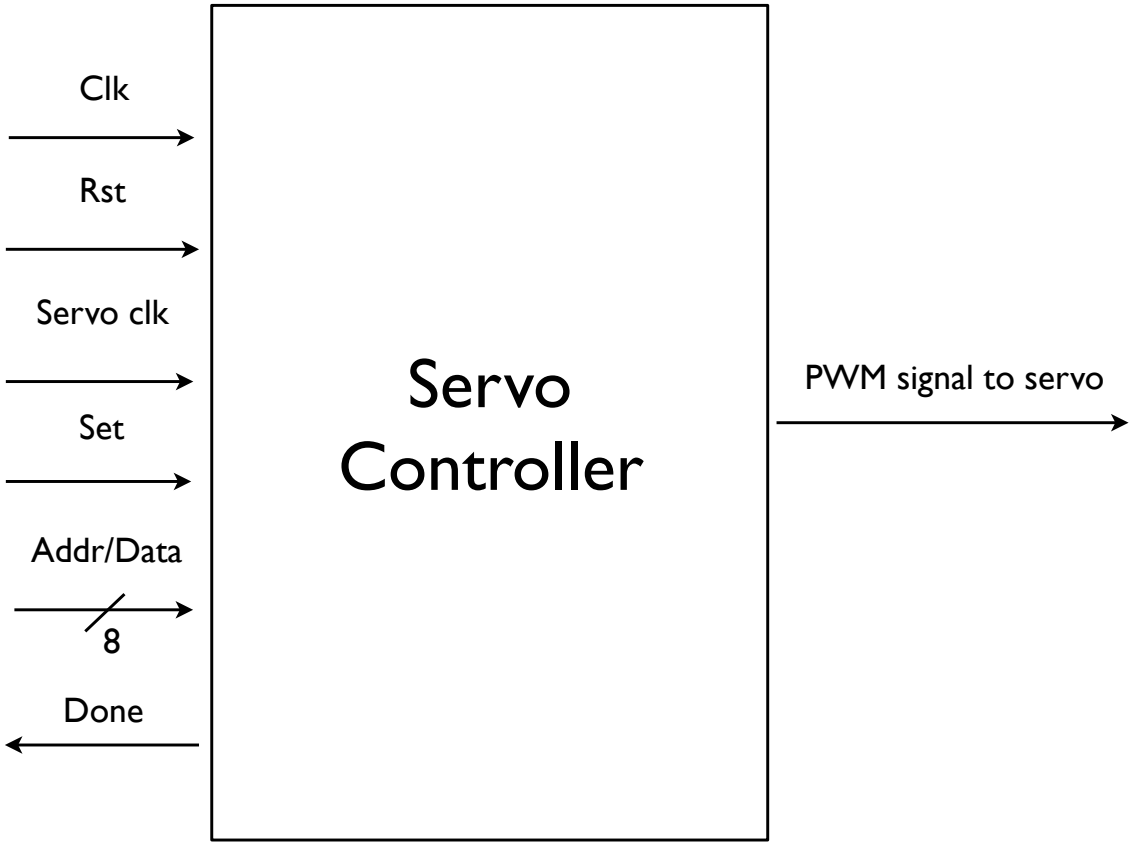
$T_{sc} = ?$

$n_{offset} = ?$

$n_{positie} = ?$



Servo controller ports



Signal to and from the servo controller

- Clk : general system clock

- Rst : asynchronous reset

- Servo clk : clock used as carrier for the PWM servo signal

- Set : request signal for an update on the PWM signal

- Addr/Data : shared 8 bit bus holding the address of the servo controller and the set value for the servo

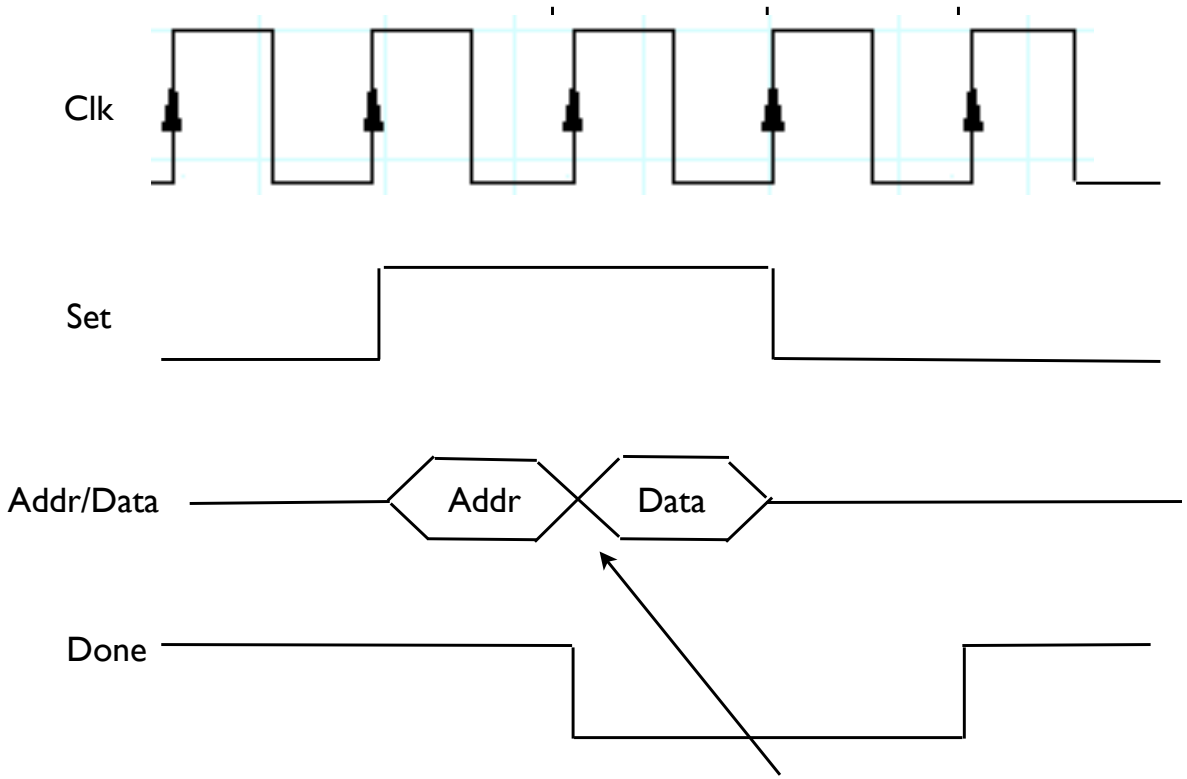
- Done: signal indicating the servo controller is ready to perform an operation

- PWM signal to servo

All signals are active high

Testability

Timing diagram



PWM set value is updated here, PWM signal itself is updated once the current PWM signal is completely built (so every 20 ms)

Other issues

your design should also cover:

Asynchronous Reset:
the servo controller returns immediately to its idle state
the servo controller output is put on zero radials and stays there until otherwise commanded by the µC

Preemption:
In normal operation the µC first puts the adress of the servo controller it wishes to communicate with on the 8 bit shared bus, then it sends the actual set value for the servo (0 left => 255 right). Meanwhile the set line is kept high for two subsequent periodes. However it can happen the µC aborts this low priority process after sending the adress information. Your system should also abort and return to the 'idle' state without ending up in somekind of a deadlock!!!

Broadcast:
Sending the adress 255 will cause all connected servo controllers to apply the set value being sent next.

Assignment:

Build the controller using VHDL fulfilling the specs as previously stated

Next session we will write a testbench to verify the design