



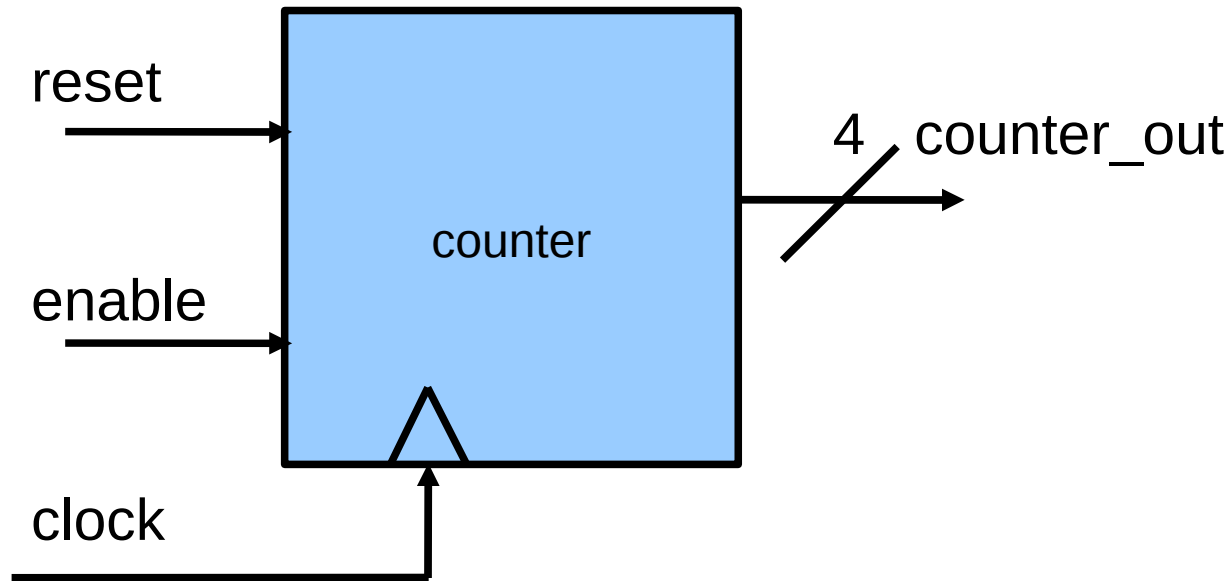
# SystemC Ex

Francesco Paterna  
Micrel Lab, DEIS, Università di Bologna  
[francesco.paterna@unibo.it](mailto:francesco.paterna@unibo.it)

# Istruzioni

1. Estrarre il pacchetto che contiene nella directory SYSTEMC la libreria SystemC precompilata per 32 e 64 bit e gli include files e nella directory Esempi\_SYSTEMC esempi di progetto
2. Editare il file SOURCEME: decommentare e commentare opportunamente le linee export SYSC\_TARGET\_ARCH=linux(64) a seconda del tipo di OS utilizzato (32 o 64 bit).
3. Eseguire: `source SOURCEME`
4. Il primo progetto esempio è contenuto all'interno della directory:  
`Esempi_SYSTEMC/Ex_COUNTER/1`
5. Il secondo progetto esempio è contenuto all'interno della directory:  
`Esempi_SYSTEMC/Ex_WRAPPER/1`

# Project 1: COUNTER



**Counter** è un modulo SystemC che incrementa un valore prodotto in uscita ad ogni colpo di clock. In ingresso sono presenti un segnale di reset per azzerare il conteggio e uno di abilitazione che tiene attivato il conteggio.

Eseguire in `Esempi_SYSTEMC/Ex_COUNTER/1`:

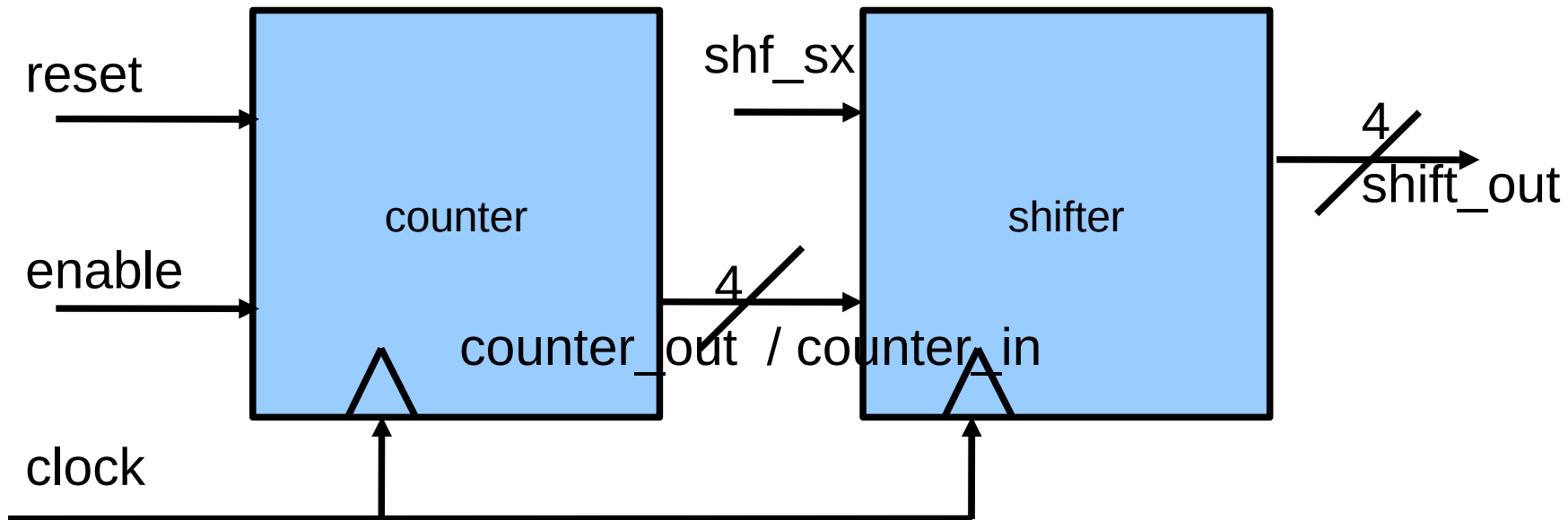
1. `make` per compilare
2. `./counter.x` per eseguire la simulazione

# Visualizza le forma d'onda

Eseguire: `gtkwave wave.vcd`

- 1) “Search” nel menu in alto
- 2) “Signal Search Regexp”
- 3) “.” Nella casella Signal Search Expression
- 4) “Select All” (in basso a sx)
- 5) “Append”
- 6) “Zoom” per visualizzare le forme d'onda

## Ex 1: counter + shifter



**Shifter** è un modulo systemc che trasla di un bit a sinistra aggiungendo uno zero a destra (molt. X 2) il contenuto in ingresso se il segnale shf\_sx è alto altrimenti trasla a destra di un bit (divisione X 2).

Aggiungere il modulo a valle del counter

# Istruzioni

1. Editare due nuovi file `shr_reg.cpp` e `shr_reg.h` dove definire il modulo e il processo associato
2. Aggiungere `shr_reg.cpp` in `CPPSRCS` del `Makefile`
3. Aggiungere in `main.cpp`:

```
#include shr_reg.h
```

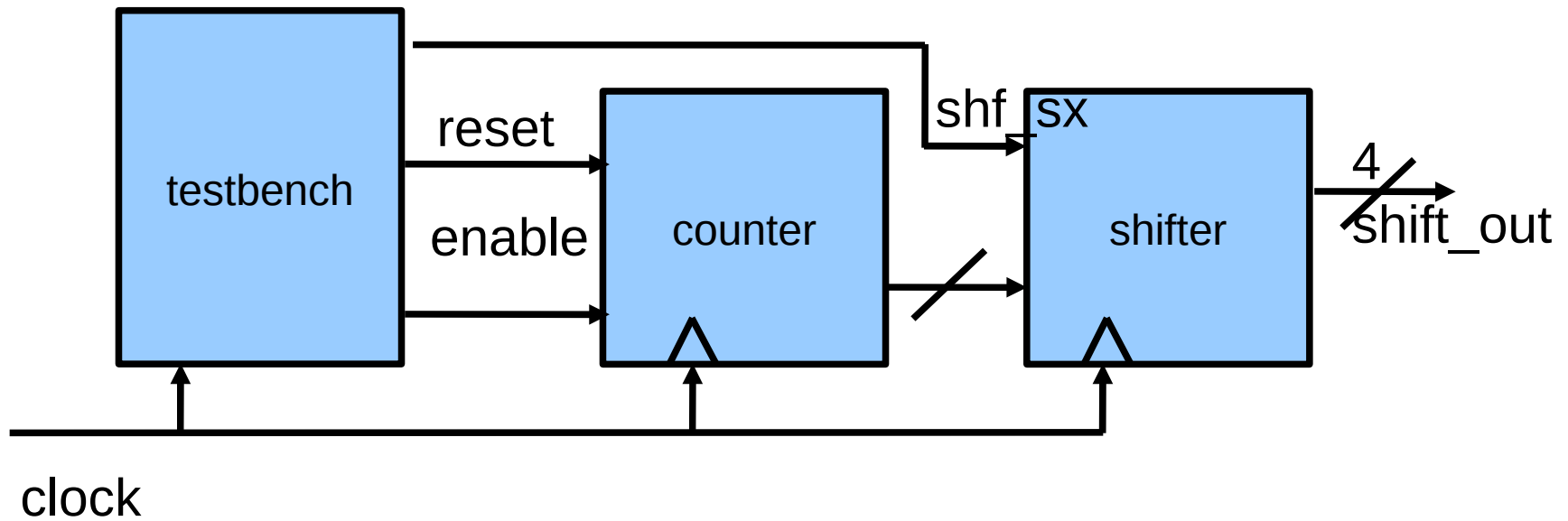
E istanziare e connettere un modulo `shr_reg`.
4. compilare con `make`, eseguire con `./counter.x` e visualizzare le forme d'onda con `gtkwave`

## suggerimenti:

- usate l'operatore `<<` per lo shift a sinistra e `>>` per lo shift a destra

**TEMPO 20 MINUTI!!!**

## Ex 2: testbench for counter + shifter



**testbench** genera il 1° ciclo di clock il reset del counter, poi per 10 cicli attiva il counter e determina lo shift a sinistra, dopodichè resetta per un ciclo nuovamente il counter prima di attivare per ulteriori 10 cicli il conteggio seguito da uno shift a destra

Aggiungere il modulo testbench per il pilotaggio del sistema.

# Istruzioni

1. Editare due nuovi file `testbench.cpp` e `testbench.h` dove definire il modulo e il processo associato
2. Aggiungere `testbench.cpp` in `CPPSRCS` del `Makefile`
3. Aggiungere in `main.cpp`:  

```
#include testbench.h
```

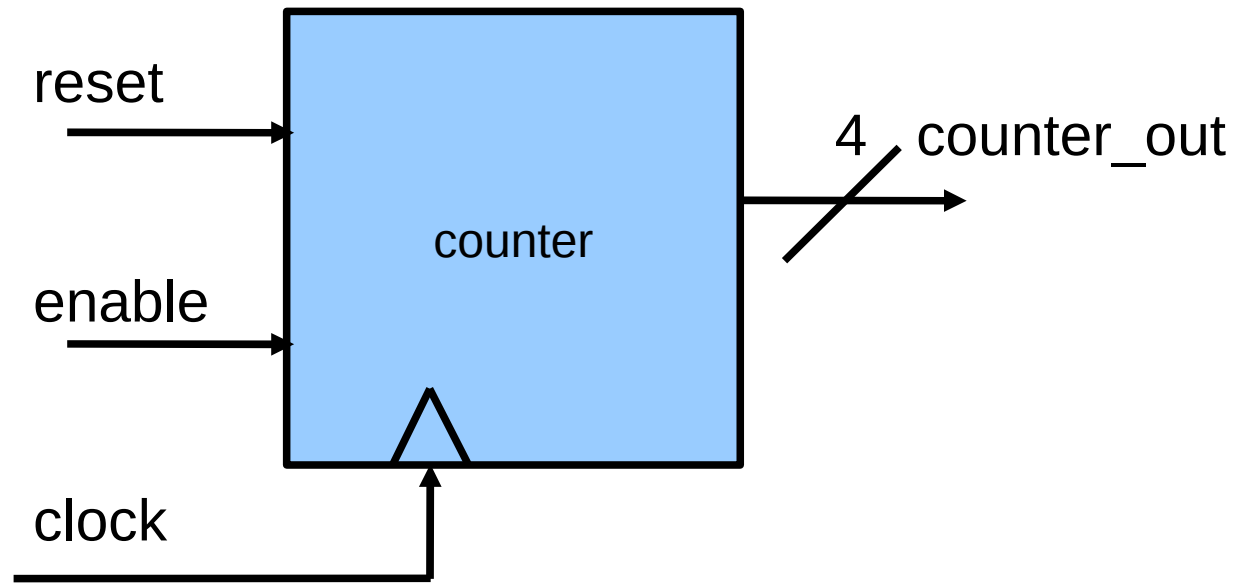
E istanziare e connettere un modulo `testbench`.
4. compilare con `make`, eseguire con `./counter.x` e visualizzare le forme d'onda con `gtkwave`

## suggerimenti:

- utilizzare un processo `SC_CTHREAD`

**TEMPO 20 MINUTI!!!**

## Ex 3: counter



Modificare counter.cpp e counter.h in modo che il processo sia SC\_METHOD anzichè SC\_THREAD

### suggerimenti:

- aggiungere come dato membro al modulo counter:

```
sc_uint<4> count_temp;
```

**TEMPO 20 MINUTI!!!**

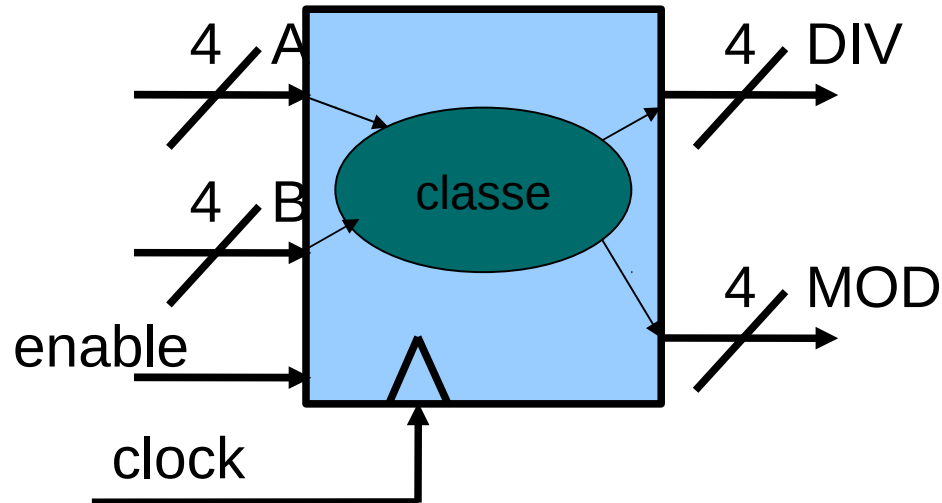
# Project 2: WRAPPER

**Classe** è una classe C++ che ha 2 dati membri. E' possibile settarli mediante costruttore con argomenti al momento della creazione dell'oggetto oppure mediante le funzioni **set**. La classe restituisce la divisione intera e il resto mediante le rispettive funzioni membro.

Eseguire in `Esempi_SYSTEMC/Ex_WRAPPER/1`:

1. `source compile` per compilare
2. `./classe "v1" "v2"` per eseguire la simulazione

## Ex 1: wrapper SystemC di *classe*



**Wrapper** è un modulo SystemC con 2 porte ingresso dati a 4 bit, 2 porte uscita dati a 4 bit, un ingresso per il clock e uno di abilitazione.

Se abilitato ad ogni istante di clock vengono campionati gli ingressi A e B e viene calcolate le uscite DIV (divisione intera) e MOD (resto).

### **suggerimenti:**

- fornire il modulo SystemC di un dato membro che è istanza di *classe*

**TEMPO 20 MINUTI!!!**



# SOLUZIONI:

## PROJECT 1:

Ex. 1 in Esempi\_SYSTEMC/Ex\_COUNTER/2

Ex. 2 in Esempi\_SYSTEMC/Ex\_COUNTER/3

Ex. 3 in Esempi\_SYSTEMC/Ex\_COUNTER/4

## PROJECT 2:

Ex. 1 in Esempi\_SYSTEMC/Ex\_WRAPPER/2