

19/02/20

# TIMEIRB

Projet de modélisation et langage système



David DEVANT  
Quentin GALES  
Victor TEXIER

Aurélien TROMPAT  
Ophélie AMOROSI  
Guillaume CHENE

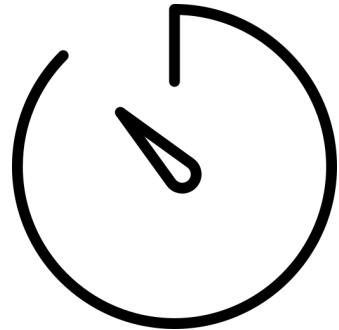
# Sommaire

- Introduction
  - Timer initial et timer implémenté
  - Fonctionnalités implémentées
  - Partie “Timer”
  - Partie “Channels”
  - Partie “Testbench”
- Conclusion

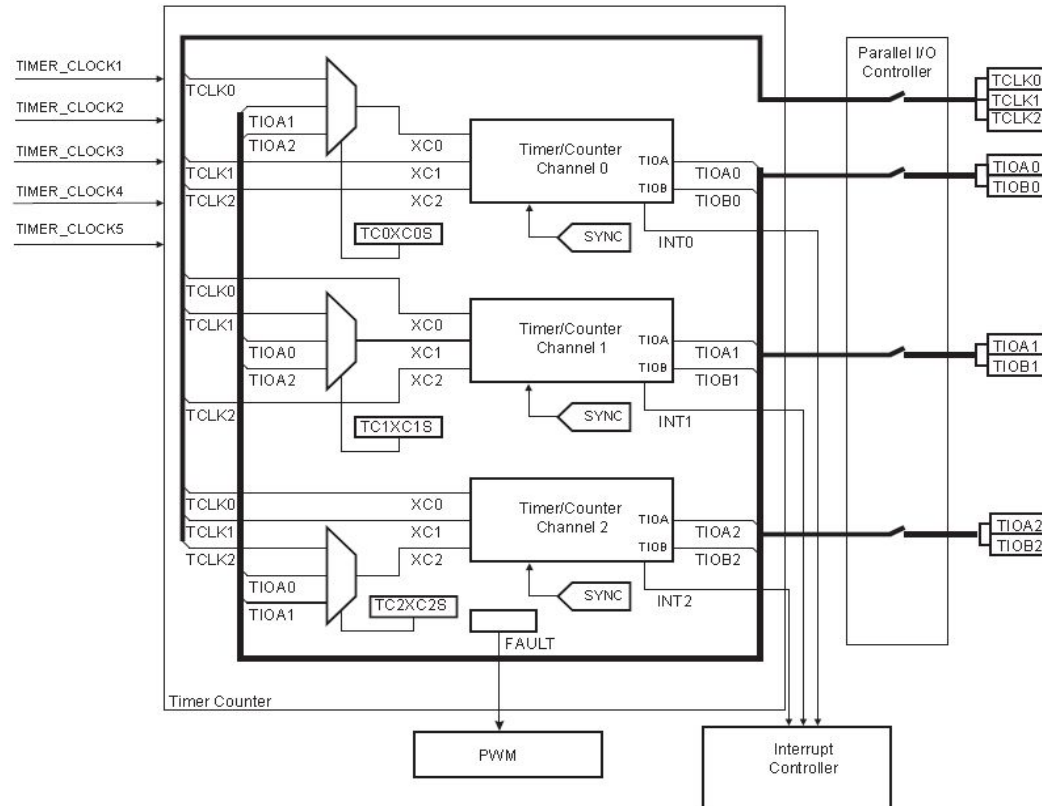


# Introduction

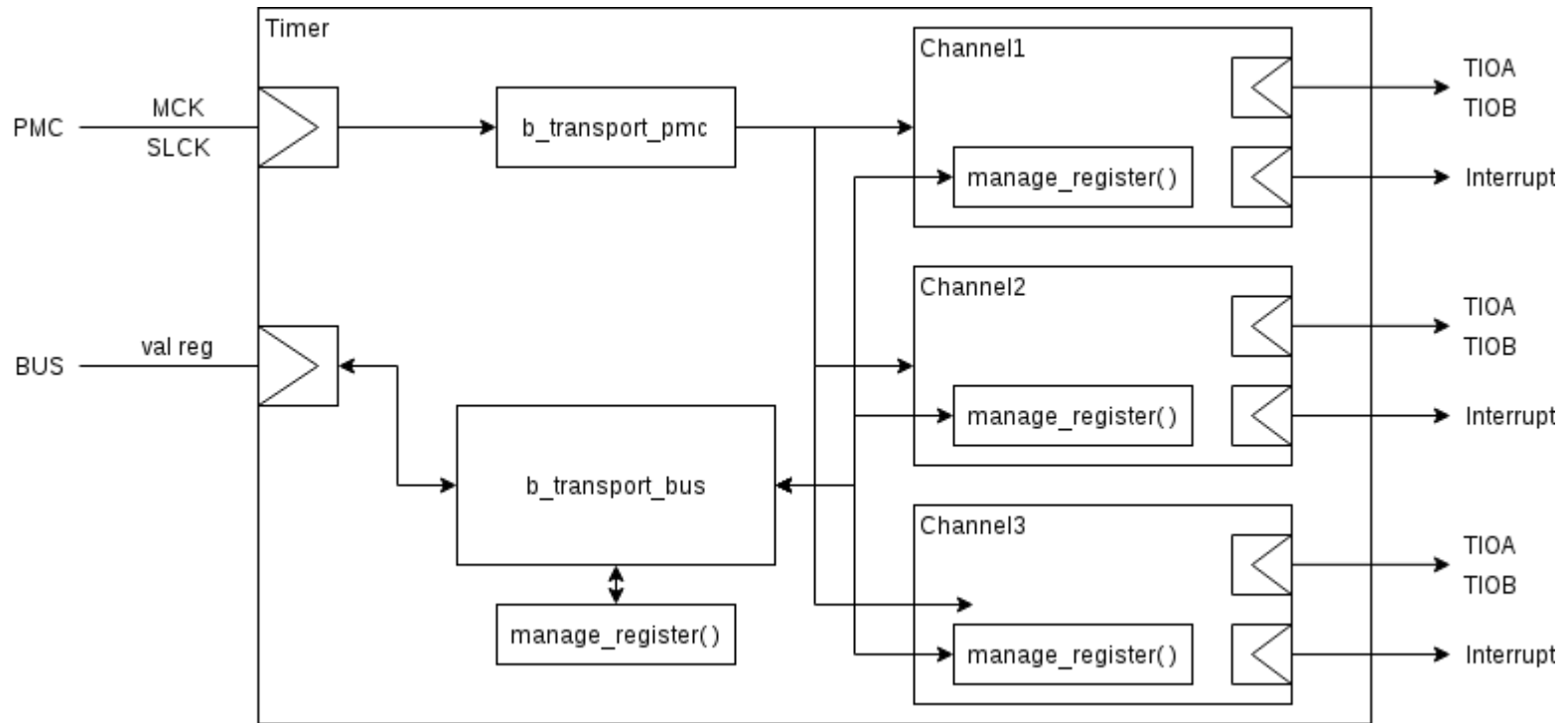
- 1 Timer Counter (duplicable)
  - 3 TC Channels identiques (32 bits)
- Fonctionnalités principales
  - Mesure en fréquence
  - Comptage d'événements
  - Mesure d'intervalles
  - Génération d'impulsions
  - Temporisation
  - PWM
- Interruptions dans les channels pour le CPU
- Quadrature decoder (filtrage, décodage signaux quadratiques ...)
  - Lecture position et vitesse d'un moteur



# Timer initial



# Timer implémenté



# Fonctionnalités implémentées

Fonctionnalité		oui	non	Commentaire
Compteur 32 bits		x		Channel -> TC Counter Value Register
Clock Selection		x		Int/Ext Clk -> TC Channel Mode Register
Clock Control		x		EN/DIS Clk -> TC Channel Control Register
Operating Mode (TC_CMR)	Capture Mode		x	TIOA TIOB: inputs
	Waveform Mode (00 et 10)	x		TIOA : output TIOB: output or external trigger
Trigger			x	
Quadrature Decoder			x	
2-bit gray up/down counter			x	
Fault Mode			x	

# Timer (TOP block)

- Réception clocks du PMC, communication de celles-ci aux channels via une méthode et print (struct pmc\_data)
- Réception données du BUS
  - Si pour channel => offset de l'adresse et envoi
  - Si pour timer lui-même => traitement
- `manage_register(uint8_t cmd, uint32_t address, uint32_t *pData)`
  - commande synchronisation des channels
  - write protection via mot de passe (pour certains registres)

# Channels

- `manage_register`(commande, adresse, pointeur de données)
  - gestion permission
  - sauvegarde des écritures en local
  - déclenchement des événements
- Gestion de la clock
  - mise à jour du compteur (valeur, overflow, rst, etc)
  - mise à jour de la fréquence / rapport cyclique
- Ecriture sur les sockets de TIOA et TIOB (sortie uniquement en waveform mode)
- Gestion des interruptions



# Interruptions

- Utilisation de `sc_event` :
  - `sc_method` avec un `sc_event` dans sa liste de sensibilité
  - Calcul du temps de la prochaine interruption et programme une notification du `sc_event`
  - Déclenchement de l'interruption par socket (channel -> testBench)
  - Mise à jour des différents registres et reprogrammation de la prochaine interruption
- Simulation de 10 secondes

# Testbench

## Objectif : Valider le fonctionnement du Timer

Tests réalisés :

- Configuration du timer
  - Clock
  - Waveform mode
  - Registres A, B et C
- Adresses du timer
  - Ecritures
  - Tests de protection
- Protections d'écriture sur le registre BMR
  - Activation et désactivation de la protection d'écriture
  - Tests d'écriture
- Interruptions dans le Timer et Channels
  - Lectures/écritures dans les registres
  - Activation et désactivation des interruptions
- Ecriture dans les registres A, B et C
  - Lecture/Ecriture
  - Protection d'écriture
- Mise à jour du compteur
  - Incréméntation par écriture
  - Lecture

# Organisation

- 1) Obtenir une chaîne complète
- 2) Développer les interfaces (Prototype des méthodes)
- 3) Communication entre les équipes “Channel” et “Timer”
- 4) Tests unitaires
- 5) Développement isolé du Testbench
- 6) Test d'intégration de la chaîne totale

Les points essentiels:

- Modularité du code
- Pas de commit qui ne compile pas
- Un makefile qui gère tous les systèmes

# Conclusion

- Prise en main de TLM2.0, SystemC et C++
- Mener un projet à 6 sur Git (utilisation de branche)
- Lecture de documentation pour isoler les fonctions utiles

## Amélioration

- Implémentation du mode capture
- Utilisation de socket Multi (TIO, Interruption)

**DevDevDavid/TimEirb**



Merci de votre attention

# Démo

## > BEGIN TIO A/B

Disabling write protection...

Resetting timer...

Setting thresholds A/B/C...

Activating clocks...

PMC to TIMER: new mck = 2000 Hz new slck = 32000 Hz

New TIO value for channel 0

TIOA Freq = 2                      TIOB Freq = 2

TIOA Duty = 4000                  TIOB Duty = 8000

Got interruption CPAS on channel 0

Got interruption CPBS on channel 0

Got interruption CPCS on channel 0

## > TIO A/B: PASSED