Concept de capture pour dater un évènement, mesurer une durée

Quand on veut dater un évènement ou mesurer la durée entre deux évènements, on peut procéder ainsi :

1 : au niveau hardware, on transforme cet évènement quel qu'il soit, en un changement d'état logique compatible avec l'arm7

-- l'outil hardware classique pour rendre ce signal propre est
soit une porte alimentée en 0 / 3V3,
soit un comparateur dont on n'oubliera pas de mettre la PULL UP à 3V3 sur sa sortie collecteur ouvert
-- il faudra vérifier que l'arm7 voit bien ce changement d'état, et que cela ne l'empeche pas de démarrer
(rappel P2.10 = EINT0 à niveau bas au démarrage => on démarre le BOOTLOADER et non notre code)

2: au niveau software, on doit prendre la photo de la valeur d'un timer qui compte de manière à dater

-- soit en capture manuelle : on part en IT (ITEXT) et on lit la valeur du Timer
-- soit en capture automatique : on utilise une entrée CAPx.y d'un timer pour que la CAPTURE se fasse automatiquement
il faut alors configurer un PINSELz pour que la patte soit configurée en tant que CAPx.y,
et configurer le comportement attendu (front choisi, IT...) llors de la capture : TxCCR
voir page suivante

3: dans le cas où l'on veut mesurer la durée entre deux évènements, il faut procéder à une différence de deux datations.

-- soit on se débrouille au moment de la capture pour remettre le timer à 0 lors d'une interruption
(ce ne peut pas être automatique malheureusement contrairement aux évènements de MATCH)
-- soit on fait la différence entre deux valeurs sans remise à zéro, mais il faut alors tester si la valeur capturée lors du second évènement est bien supérieure à celle du premier (sinon on a affaire à un rebouclage du timer, et il faut en tenir compte en ajoutant la valeur de rebouclage à cette différence négative de manière erronée)

Ces techniques peuvent être utilisées pour :

mesurer la durée écoulée entre une émission et la réception d'un écho

analyser un motif d'un protocole (on classera alors la durée mesurée en plusieurs cas : MotifA, MotifB, MotifC ...., Erreur)

Toujours Faire attention à ce que l'information à récupérer ne soit pas écrasée entre temps (entre la capture et la récupération)
par une autre capture qui arriverait rapidement. On pensera au comparateur à double seuil, pour éviter un rebond du signal par exemple

TIMER

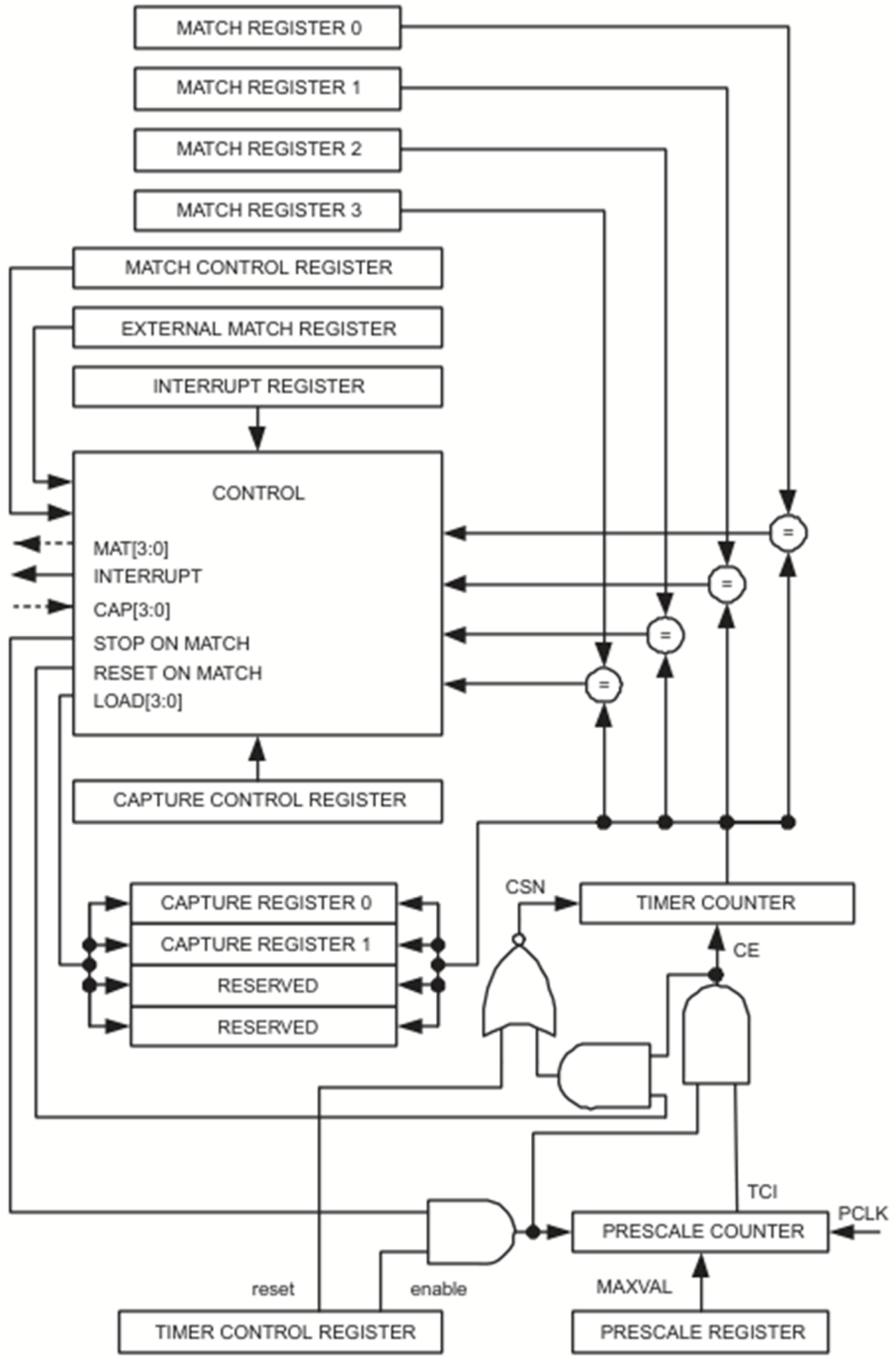
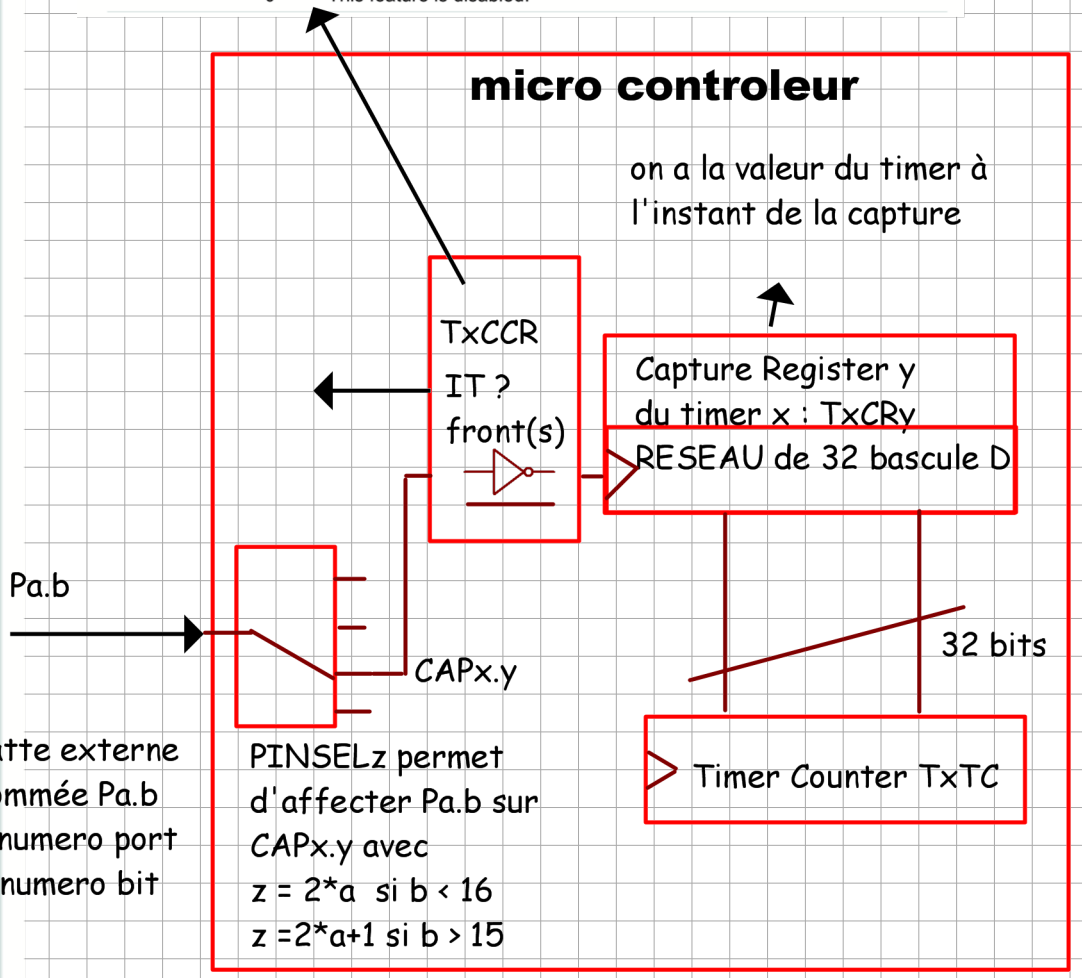


Fig 127. Timer block diagram

PRINCIPE d'une capture

Table 477: Capture Control Register (T[0/1/2/3]CCR - addresses 0xE000 4028, 0xE000 8020, 0xE007 0028, 0xE007 4028) bit description

Bit	Symbol	Value	Description	Reset Value
0	CAP0RE	1	Capture on CAPn.0 rising edge: a sequence of 0 then 1 on CAPn.0 will cause CR0 to be loaded with the contents of TC.	0
		0	This feature is disabled.	
1	CAP0FE	1	Capture on CAPn.0 falling edge: a sequence of 1 then 0 on CAPn.0 will cause CR0 to be loaded with the contents of TC.	0
		0	This feature is disabled.	
2	CAP0I	1	Interrupt on CAPn.0 event: a CR0 load due to a CAPn.0 event will generate an interrupt.	0
		0	This feature is disabled.	



x numero timer (0 1 2 3)
y numero capture (0 1)