```
void TrigThreading(float BaseSpeedRPS)
{
	register SPINDLE *S=&Spindle;

	if (BaseSpeedRPS < MIN_RPS) BaseSpeedRPS = MIN_RPS;

	S->InvBaseSpeedRPS = 1.0f/BaseSpeedRPS;
	S->InvBaseSpeedRPSK1 = S->InvBaseSpeedRPS * (1.0 - S->K);

	CS0_t = CS0_TimeBase = CS0_TimeBaseDelta = S->LastCSTime = 0.0;
	CoordSystem0 = LastCoordSystem0;

	// compute where to start
	S->StartPosition = (int)(S->Position) + 2;

	// Initialize the Filter to the current unfiltered without any Lead
	S->AdjTimeFilt = (S->Position - S->StartPosition) * S->InvBaseSpeedRPS;

	// Activate last after everything is set up
	S->ThreadingActive = TRUE;
}
```

```
    6364 0000169c           _TrigThreading:
    6365                    ;** --------------------------------------------------------------------------
*
    6366 0000169c 073CD4F4            STW     .D2T1   A14,*SP--(24)     ; |2734|
    6367 000016a0 05BC62F6            STW     .D2T2   B11,*+SP(12)      ; |2734|
    6368 000016a4 063C82F6            STW     .D2T2   B12,*+SP(16)      ; |2734|
    6369 000016a8 06BCA2F6            STW     .D2T2   B13,*+SP(20)      ; |2734|
    6370
    6371 000016ac 070C1059            MV      .L1X    B3,A14            ; |2734|
    6372 000016b0 053C42F6  ||        STW     .D2T2   B10,*+SP(8)       ; |2734|
    6373
    6374                    $C$DW$294       .dwtag  DW_TAG_formal_parameter, DW_AT_name("BaseSpeedRPS")
    6375                            .dwattr $C$DW$294, DW_AT_TI_symbol_name("_BaseSpeedRPS")
    6376                            .dwattr $C$DW$294, DW_AT_type(*$C$DW$T$16)
    6377                            .dwattr $C$DW$294, DW_AT_location[DW_OP_reg4]
    6378
    6379 000016b4 0210105B            MV      .L2X    A4,B4             ; |2734|
ÀMS320C6x COFF Assembler PC v5.3.0 Sat Apr 12 20:31:03 2014

Tools Copyright (c) 1996-2005 Texas Instruments Incorporated
fast.asm                                                        PAGE  117

    6380 000016b8 01800028! ||        MVKL    .S1     __divf,A3
    6381
    6382 000016c0 031000A3            SPDP    .S2     B4,B7:B6          ; |2737|
    6383 000016c4 01800068! ||        MVKH    .S1     __divf,A3
    6384
    6385 000016c8 029705A8            MVKL    .S1     0x3e112e0b,A5
    6386
    6387 000016cc 000C1363            CALL    .S2X    A3                ; |2739|
    6388 000016d0 026B4AA8  ||        MVKL    .S1     0xe826d695,A4
    6389
    6390 000016d4 04382FAB            MVKL    .S2     0x3089705f,B8
    6391 000016d8 029F08E8  ||        MVKH    .S1     0x3e112e0b,A5
    6392
    6393 000016e0 018B862B+           MVKL    .S2     $C$RL98,B3        ; |2739|
    6394 000016e4 02741369  ||        MVKH    .S1     0xe826d695,A4
```

```
    6395 000016e8 028000FA  ||         ZERO    .L2     B5                ; |2739|
    6396
    6397 000016ec 00989A61             CMPLTDP .S1X    B7:B6,A5:A4,A1    ; |2737|
    6398 000016f0 029FC06A  ||         MVKH    .S2     0x3f800000,B5     ; |2739|
    6399
    6400 000016f4 041844EA             MVKH    .S2     0x3089705f,B8
    6401
    6402 00001700 8220005B     [ A1]   MV      .L2     B8,B4             ; |2737|
    6403 00001704 02141059  ||         MV      .L1X    B5,A4             ; |2739|
    6404 00001708 0180006A+ ||         MVKH    .S2     $C$RL98,B3        ; |2739|
    6405
    6406 0000170c           $C$RL98:   ; CALL OCCURS {A3}                ; |2739|
    6407                    ;** --------------------------------------------------------------------------
*
    6408 0000170c 00000000             NOP             1
    6409 00001710 01803C28-            MVKL    .S1     _Spindle,A3
    6410 00001714 01800068-            MVKH    .S1     _Spindle,A3
    6411 00001718 030C0364             LDDW    .D1T1   *A3,A7:A6         ; |2746|
    6412 0000171c 00006000             NOP             4
    6413 00001720 019CC038             DPTRUNC .L1     A7:A6,A3          ; |2746|
    6414 00001724 00004000             NOP             3
    6415 00001728 018C4058             ADD     .L1     2,A3,A3           ; |2746|
    6416 0000172c 000C0738             INTDP   .L1     A3,A1:A0          ; |2746|
    6417 00001730 00000000             NOP             1
    6418 00001734 0200602A-            MVKL    .S2     _Spindle+72,B4
    6419 00001738 0200006A-            MVKH    .S2     _Spindle+72,B4
    6420 0000173c 021003E6             LDDW    .D2T2   *B4,B5:B4         ; |2740|
    6421 00001740 0400C338             SUBDP   .L1     A7:A6,A1:A0,A9:A8 ; |2749|
    6422 00001744 00000000             NOP             1
    6423 00001748 038000F8             ZERO    .L1     A7                ; |2740|
    6424
    6425 0000174c 039FF869             MVKH    .S1     0x3ff00000,A7     ; |2740|
    6426 00001750 030000F8  ||         ZERO    .L1     A6                ; |2740|
    6427
    6428 00001754 0210D3BA             SUBDP   .L2X    A7:A6,B5:B4,B5:B4 ; |2740|
    6429 00001758 00000000             NOP             1
    6430 0000175c 0080002A!            MVKL    .S2     _CS0_TimeBase,B1
    6431 00001760 0180582A-            MVKL    .S2     _Spindle+56,B3
    6432 00001764 0500002A!            MVKL    .S2     _CS0_t,B10
    6433
    6434 00001768 031000A1             SPDP    .S1     A4,A7:A6          ; |2740|
ÀMS320C6x COFF Assembler PC v5.3.0 Sat Apr 12 20:31:03 2014

Tools Copyright (c) 1996-2005 Texas Instruments Incorporated
fast.asm                                                     PAGE  118

    6435 0000176c 0000182A  ||         MVK     .S2     48,B0
    6436
    6437 00001770 0600322A             MVK     .S2     100,B12
    6438
    6439 00001774 04189703             MPYDP   .M2X    A7:A6,B5:B4,B9:B8 ; |2740|
    6440 00001778 0080006A! ||         MVKH    .S2     _CS0_TimeBase,B1
    6441
    6442 00001780 0180006B-            MVKH    .S2     _Spindle+56,B3
    6443 00001784 030000FA  ||         ZERO    .L2     B6                ; |2742|
    6444
    6445 00001788 030402F7             STW     .D2T2   B6,*B1            ; |2742|
    6446 0000178c 0500006A! ||         MVKH    .S2     _CS0_t,B10
    6447
    6448 00001790 030424FB             ZERO    .L2     B7:B6             ; |2742|
    6449 00001794 000065E3  ||         SUB     .S2     B3,B0,B0
    6450 00001798 032802F6  ||         STW     .D2T2   B6,*B10           ; |2742|
    6451
    6452 000017a0 030C02F7             STW     .D2T2   B6,*B3            ; |2742|
    6453 000017a4 0630007B  ||         ADD     .L2     B0,B12,B12
    6454 000017a8 0100002B! ||         MVKL    .S2     _CS0_TimeBaseDelta,B2
    6455 000017ac 028000A8  ||         MVK     .S1     1,A5              ; |2752|
    6456
    6457 000017b0 0320C701             MPYDP   .M1     A7:A6,A9:A8,A7:A6 ; |2749|
```

```
    6458 000017b4 018000F9  ||         ZERO    .L1     A3                ; |2742|
    6459 000017b8 02B002F5  ||         STW     .D2T1   A5,*B12           ; |2752|
    6460 000017bc 0100006A! ||         MVKH    .S2     _CS0_TimeBaseDelta,B2
    6461
    6462 000017c0 018802F4             STW     .D2T1   A3,*B2            ; |2742|
    6463
    6464 000017c4 038C22F7             STW     .D2T2   B7,*+B3(4)        ; |2742|
    6465 000017c8 01000C28  ||         MVK     .S1     24,A2
    6466
    6467 000017cc 000002F5             STW     .D2T1   A0,*B0            ; |2746|
    6468 000017d0 028425E1  ||         ZERO    .S1     A5                ; |2742|
    6469 000017d4 010992F8  ||         SUB     .L1X    B12,A2,A2
    6470
    6471 000017e0 028822F5             STW     .D2T1   A5,*+B2(4)        ; |2742|
    6472 000017e4 0688905B  ||         ADD     .L2X    4,A2,B13
    6473 000017e8 0580002A! ||         MVKL    .S2     _LastCoordSystem0,B11
    6474
    6475 000017ec 043402F7             STW     .D2T2   B8,*B13           ; |2740|
    6476 000017f0 0580006A! ||         MVKH    .S2     _LastCoordSystem0,B11
    6477
    6478 000017f4 022C02E7             LDW     .D2T2   *B11,B4           ; |2743|
    6479 000017f8 028000FA  ||         ZERO    .L2     B5                ; |2742|
    6480
    6481 000017fc 02A822F6             STW     .D2T2   B5,*+B10(4)       ; |2742|
    6482 00001800 04B422F6             STW     .D2T2   B9,*+B13(4)       ; |2740|
    6483
    6484 00001804 028422F7             STW     .D2T2   B5,*+B1(4)        ; |2742|
    6485 00001808 04000029! ||         MVKL    .S1     _CoordSystem0,A8
    6486 0000180c 018F1059  ||         ADD     .L1X    -8,B3,A3
    6487 00001810 02080274  ||         STW     .D1T1   A4,*A2            ; |2739|
    6488
    6489 00001814 053C23E7             LDDW    .D2T2   *+SP(8),B11:B10   ; |2753|
ÀMS320C6x COFF Assembler PC v5.3.0 Sat Apr 12 20:31:03 2014

Tools Copyright (c) 1996-2005 Texas Instruments Incorporated
fast.asm                                                        PAGE  119

    6490 00001818 04000069! ||         MVKH    .S1     _CoordSystem0,A8
    6491 0000181c 030C0274  ||         STW     .D1T1   A6,*A3            ; |2749|
    6492
    6493 00001820 02200277             STW     .D1T2   B4,*A8            ; |2743|
    6494 00001824 008022F5  ||         STW     .D2T1   A1,*+B0(4)        ; |2746|
    6495 00001828 01B8105A  ||         MV      .L2X    A14,B3            ; |2753|
    6496
    6497 0000182c 000C0363             RET     .S2     B3                ; |2753|
    6498 00001830 038C2275  ||         STW     .D1T1   A7,*+A3(4)        ; |2749|
    6499 00001834 063C43E6  ||         LDDW    .D2T2   *+SP(16),B13:B12  ; |2753|
    6500
    6501 00001838 073CD2E4             LDW     .D2T1   *++SP(24),A14     ; |2753|
    6502                             .dwpsn  "fast.c",2753,1
    6503 0000183c 00006000             NOP             4
    6504                               ; BRANCH OCCURS {B3}              ; |2753|
```

Tried using a function as a "Memory Barrier" and that didn't work either. Interestingly the function was created but never called. Instead the code in the funtion was still inlined out of order.

```
void MemoryBarrier(void)
{
	// Activate last after everything is set up
	Spindle.ThreadingActive = TRUE;
}


#define MIN_RPS 1e-9
void TrigThreading(float BaseSpeedRPS)
{
	register SPINDLE *S=&Spindle;

	if (BaseSpeedRPS < MIN_RPS) BaseSpeedRPS = MIN_RPS;

	S->InvBaseSpeedRPS = 1.0f/BaseSpeedRPS;
	S->InvBaseSpeedRPSK1 = S->InvBaseSpeedRPS * (1.0 - S->K);

	CS0_t = CS0_TimeBase = CS0_TimeBaseDelta = S->LastCSTime = 0.0;
	CoordSystem0 = LastCoordSystem0;

	// compute where to start
	S->StartPosition = (int)(S->Position) + 2;

	// Initialize the Filter to the current unfiltered without any Lead
	S->AdjTimeFilt = (S->Position - S->StartPosition) * S->InvBaseSpeedRPS;

	// Activate last after everything is set up
	MemoryBarrier();
}
```

This was suggested. Actually made a smaller window but still not correct

```
inline void force_write(int *ptr, int value)
{
    *(volatile int *)ptr = value;
}

#define MIN_RPS 1e-9
void TrigThreading(float BaseSpeedRPS)
{
      register SPINDLE *S=&Spindle;

      if (BaseSpeedRPS < MIN_RPS) BaseSpeedRPS = MIN_RPS;

      S->InvBaseSpeedRPS = 1.0f/BaseSpeedRPS;
      S->InvBaseSpeedRPSK1 = S->InvBaseSpeedRPS * (1.0 - S->K);

      CS0_t = CS0_TimeBase = CS0_TimeBaseDelta = S->LastCSTime = 0.0;
      CoordSystem0 = LastCoordSystem0;

      // compute where to start
      S->StartPosition = (int)(S->Position) + 2;

      // Initialize the Filter to the current unfiltered without any Lead
      S->AdjTimeFilt = (S->Position - S->StartPosition) * S->InvBaseSpeedRPS;

      // Activate last after everything is set up
      force_write(&Spindle.ThreadingActive,TRUE);
}
```

I think this will be my solution. I was worried that it might not put the WaitNextTimeSlice at the beginning but it did (not 100% sure it always will). Although it did in-line all the code! Which should be ok.

```
// WaitNextTimeSlice - wait until a thread's new time slice begins

double WaitNextTimeSlice(void)
{
	register double tick=ServoTick;

	while (tick==ServoTick) ;

	return Time_sec();
}




void TrigThreading(float BaseSpeedRPS)
{
	register SPINDLE *S=&Spindle;

	WaitNextTimeSlice();

	if (BaseSpeedRPS < MIN_RPS) BaseSpeedRPS = MIN_RPS;

	S->InvBaseSpeedRPS = 1.0f/BaseSpeedRPS;
	S->InvBaseSpeedRPSK1 = S->InvBaseSpeedRPS * (1.0 - S->K);

	CS0_t = CS0_TimeBase = CS0_TimeBaseDelta = S->LastCSTime = 0.0;
	CoordSystem0 = LastCoordSystem0;

	// compute where to start
	S->StartPosition = (int)(S->Position) + 2;

	// Initialize the Filter to the current unfiltered without any Lead
	S->AdjTimeFilt = (S->Position - S->StartPosition) * S->InvBaseSpeedRPS;

	// Activate last after everything is set up
	S->ThreadingActive = TRUE;
}
```