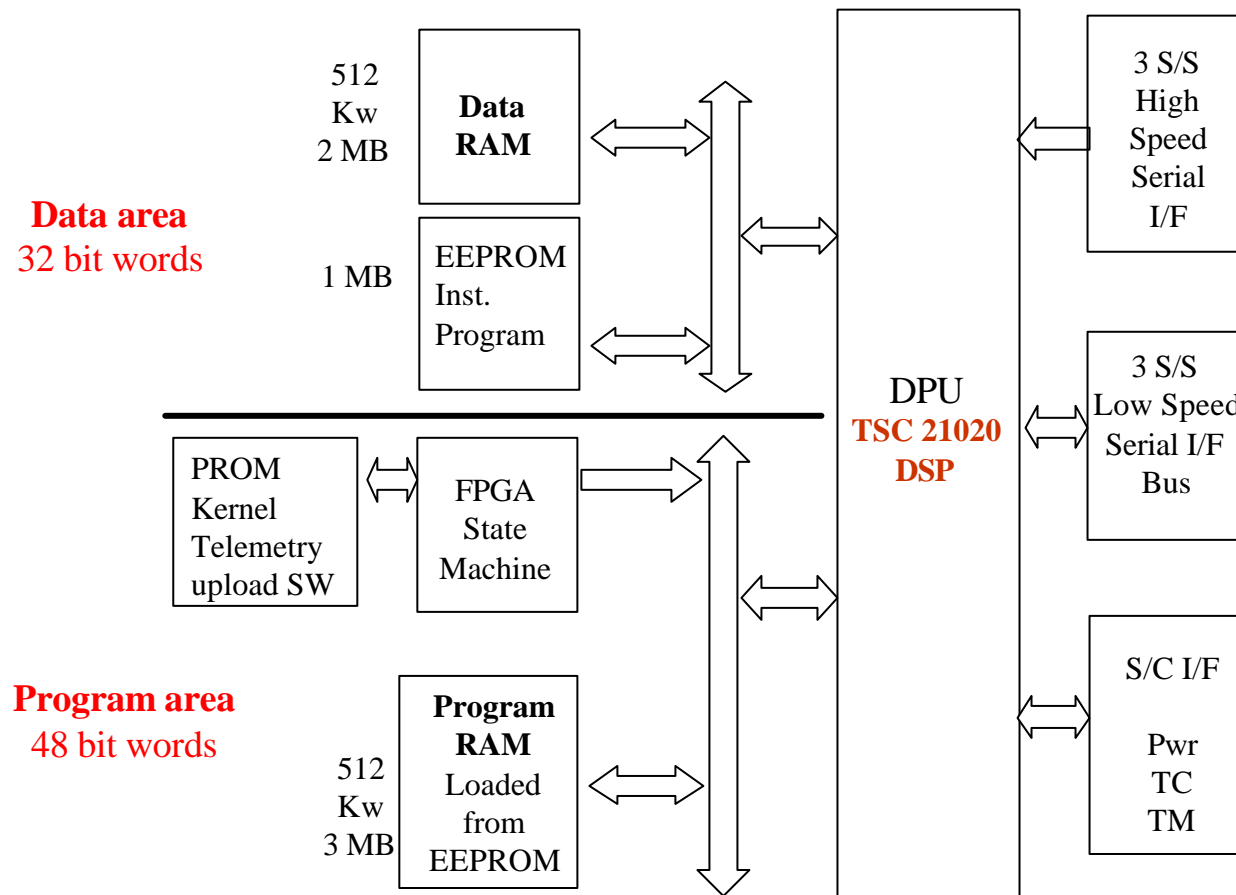
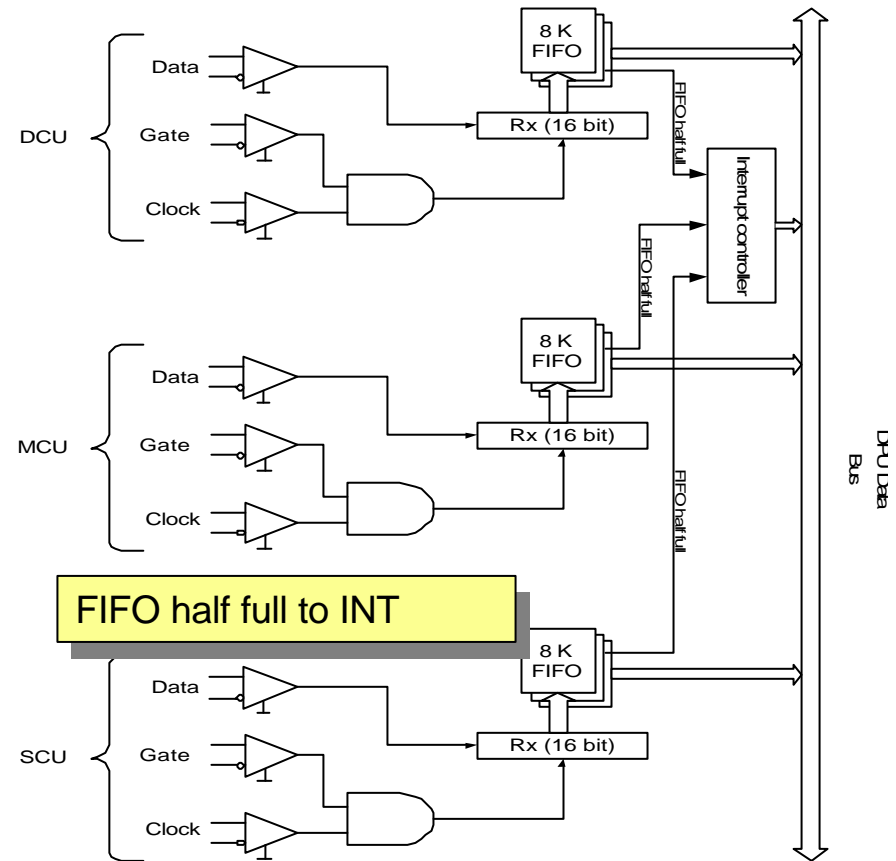
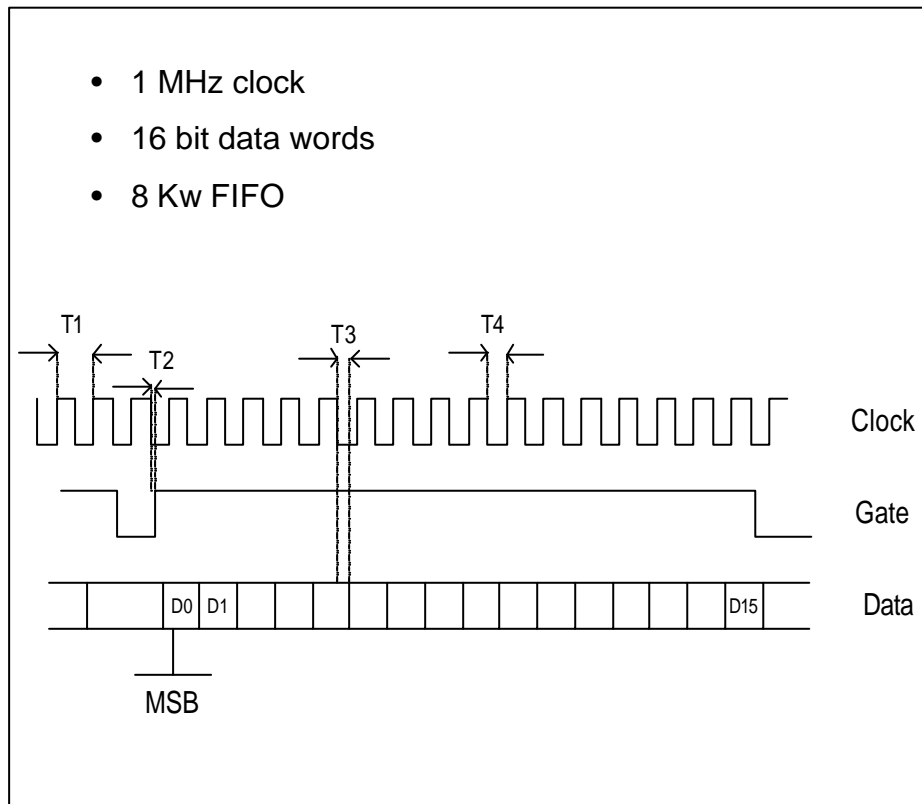


DPU/ICU and OBS

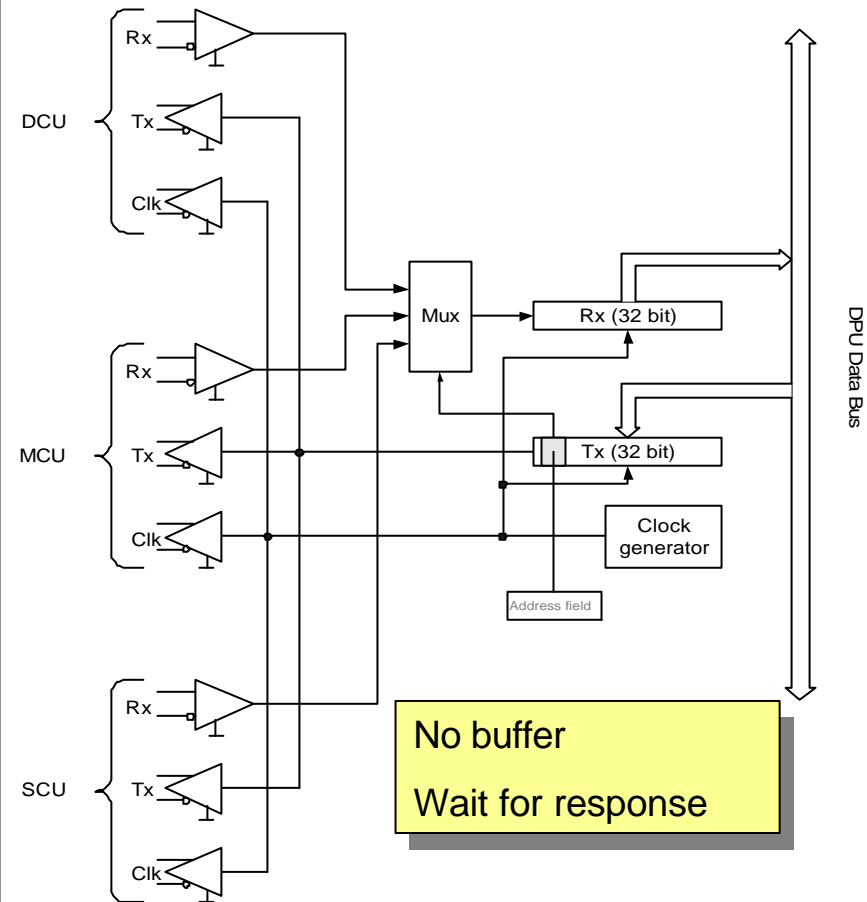
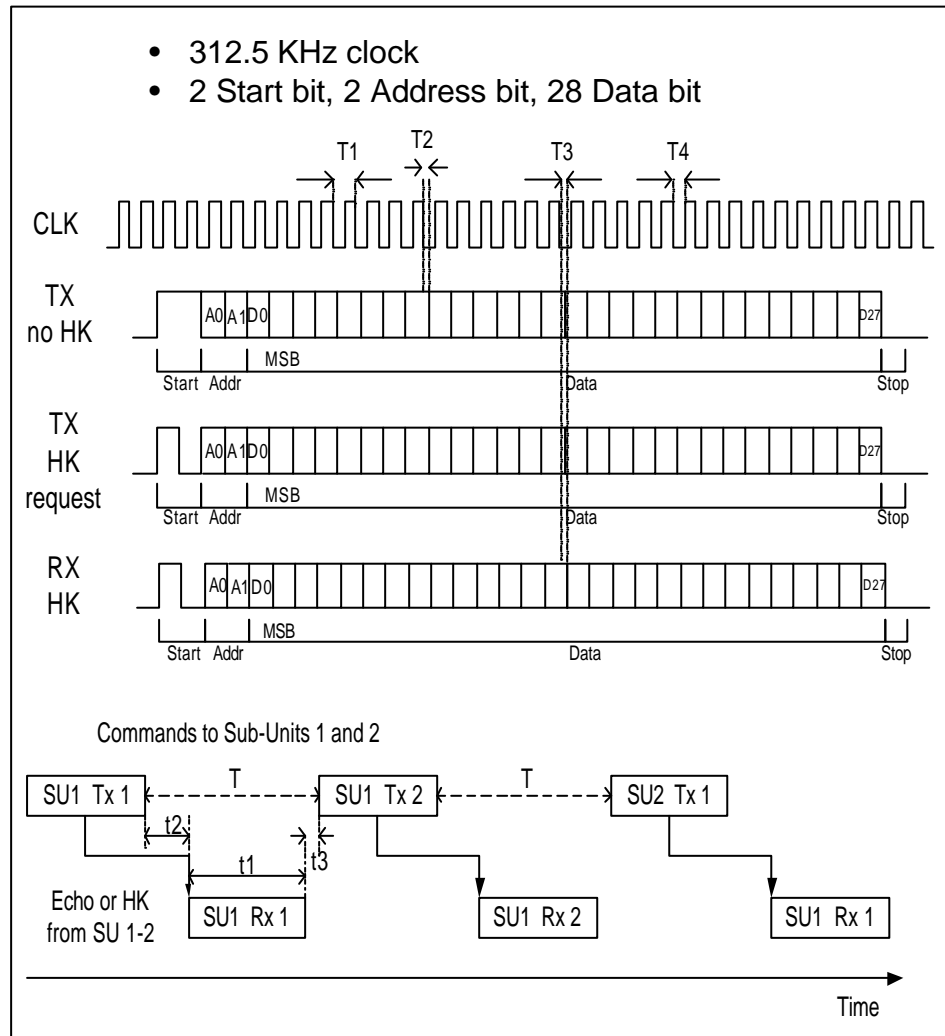
DPU CPU memory organization



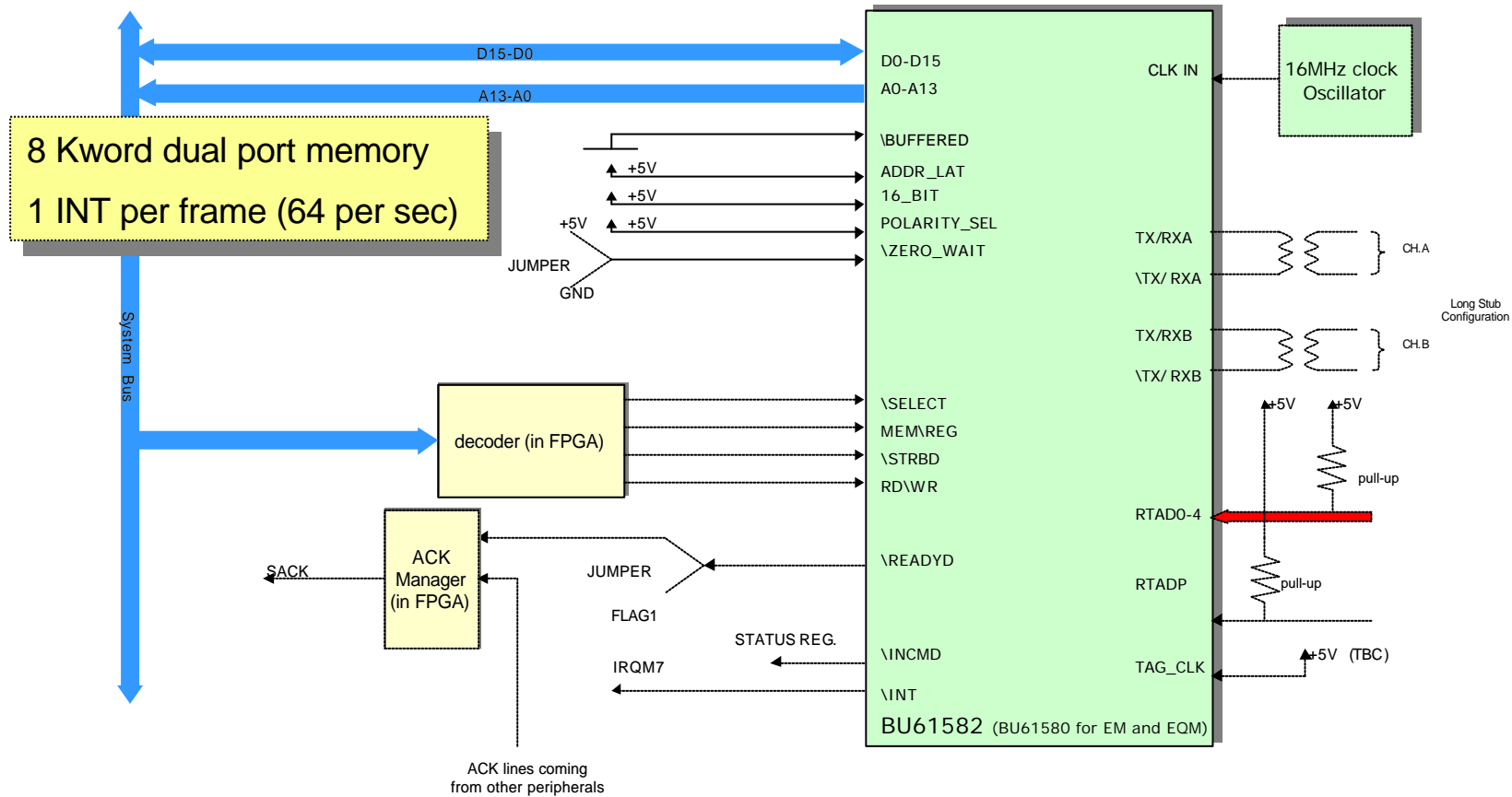
High Speed Interface



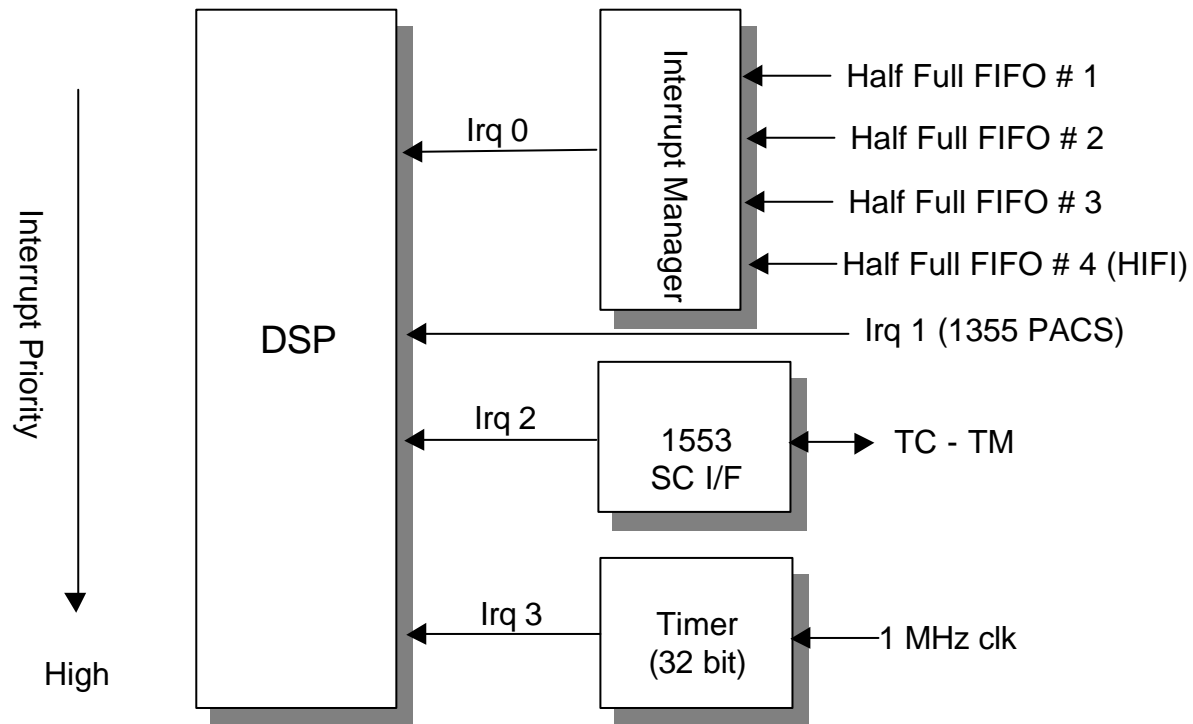
Low Speed Interface



MIL-STD-1553 interface



Interrupt generation

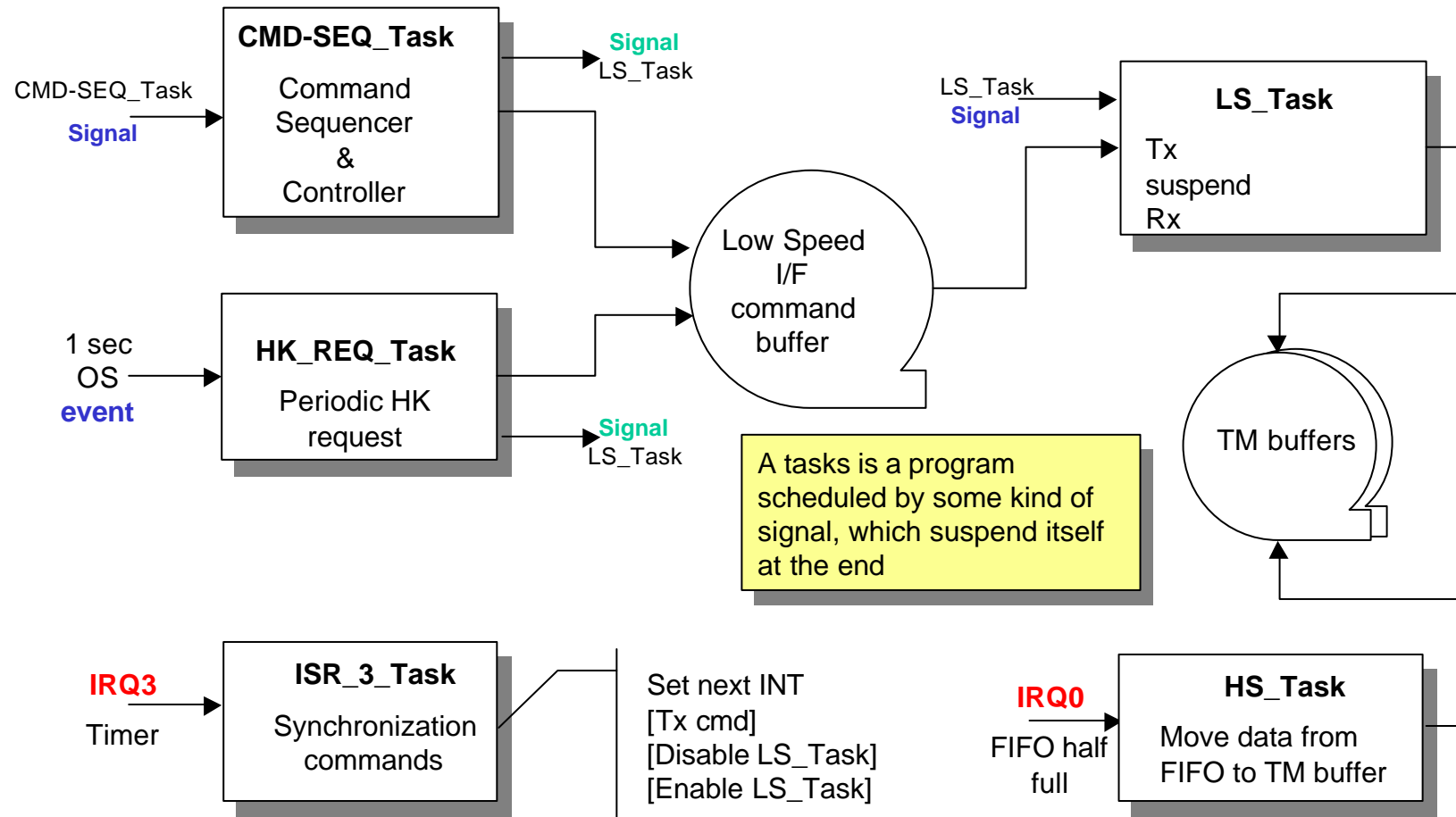


OBS Environment

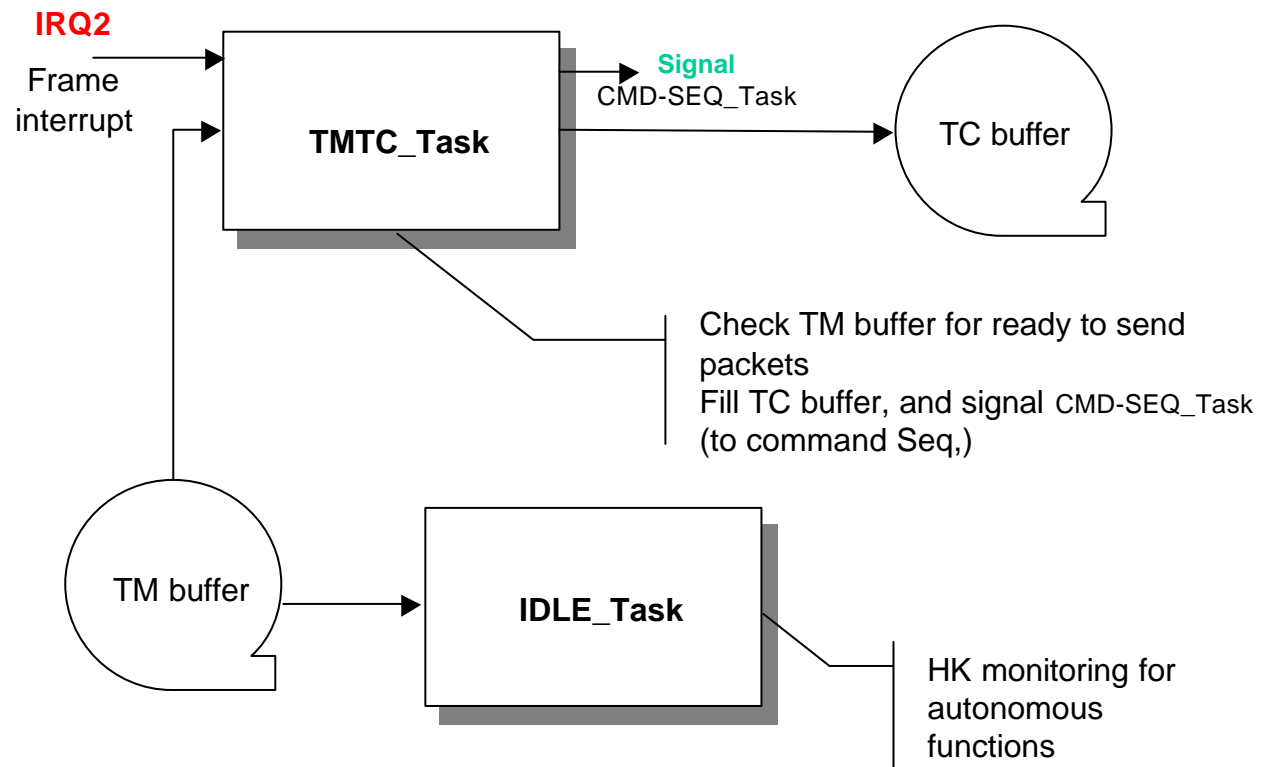
IFSI	DPU OBS	S/W Patching
CGS	1553 Drivers	Boot S/W
Eonic	Virtuoso O.S.	

- Eonic “Virtuoso” multitask O.S.
- Inter-task communication via circular buffers in global memory (fastest)
- Inter-task synchronization via semaphores and mutex
- Task switching overhead ~ 100 us

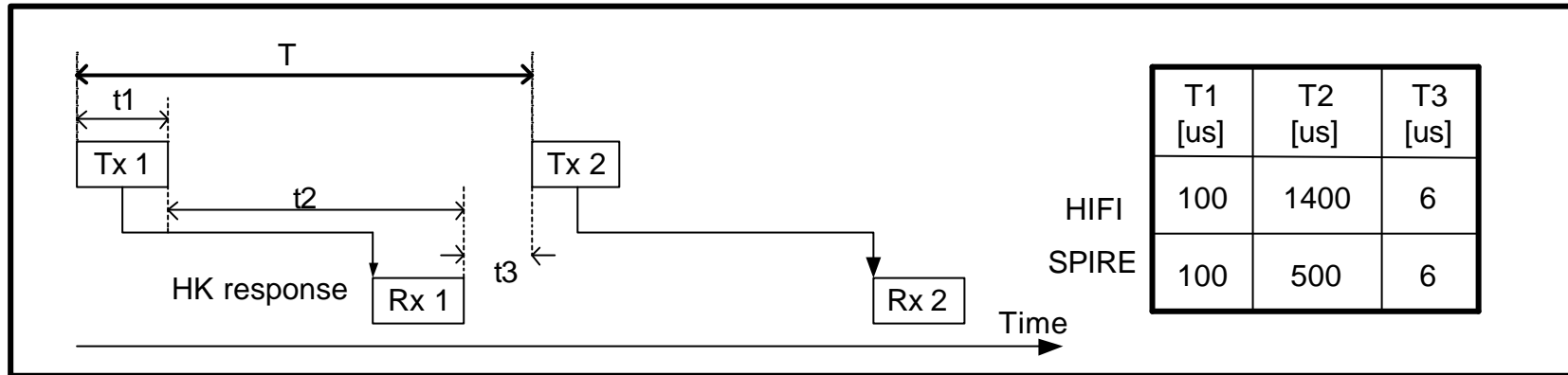
OBS Tasks



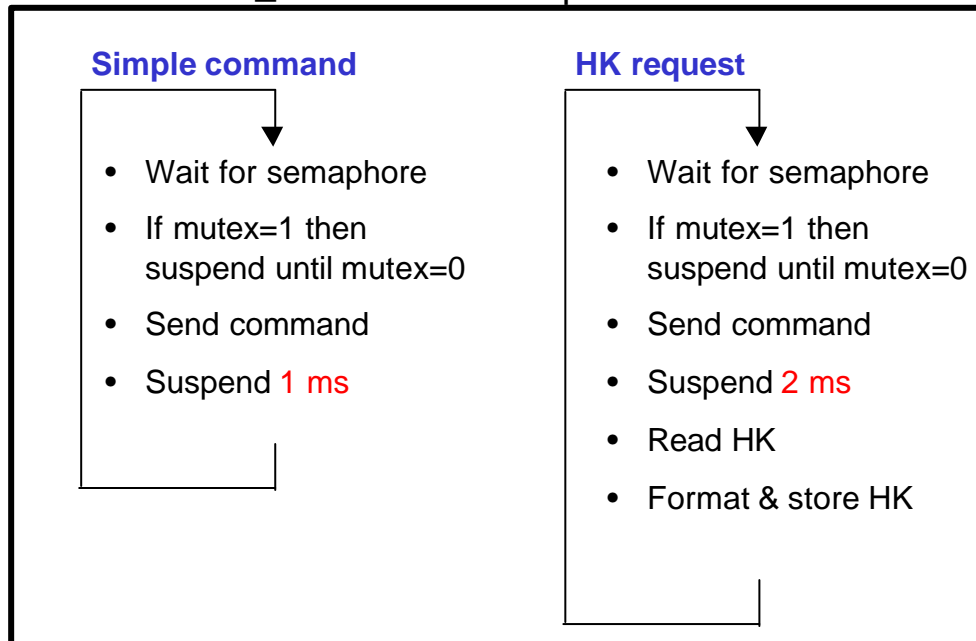
OBS Tasks



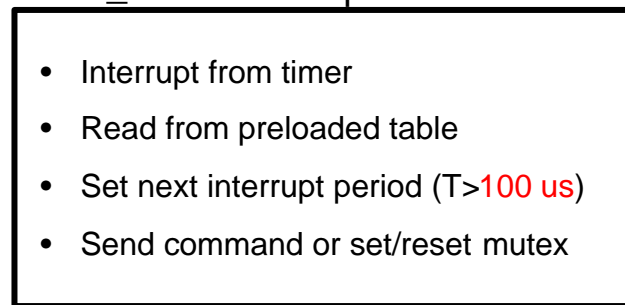
Low Speed Interfaces Timing



LS_Task: 1 ms time quantization

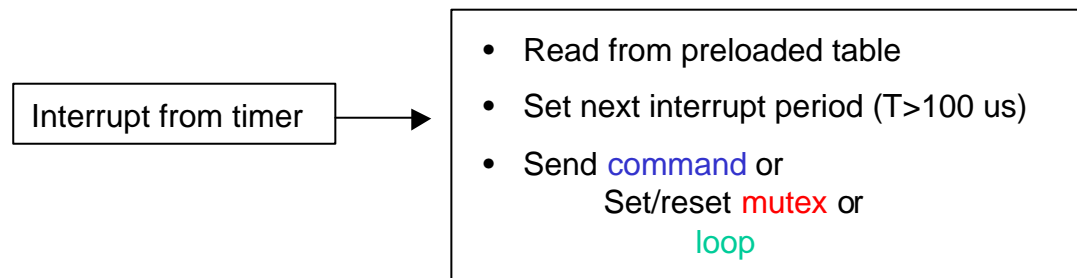


ISR_3: 1 us time quantization



SYNC_Task (ISR 3)

For every down_counter timer generated INT 3, a new row in the ISR 3 exec_matrix is executed. Down_counter clock period $t = 1 \text{ us}$



exec_matrix		
T [us] next INT	CMD	LS_Task Mutex
2000	Z	1
250	0x8765	Z
40000	0xBABA	Z
3000	Z	0
2000	Z	1

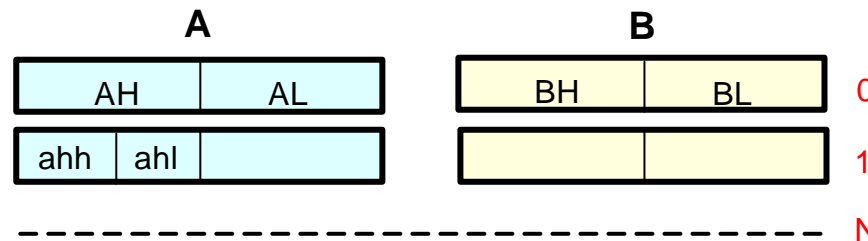
Incrementing
pointer →

```

    ...
    Set LS_mtx
        xx ms "suspend"
    Out CMD=0x8765
        2000 us "suspend"
    Out CMD=0xBABA
        250 us "suspend"
    Reset LS_mtx
        40 ms "suspend"
    Set LS_mtx
        3000 us "suspend"
    ....
    ....
  
```

SYNC_Task (ISR 3) – exec_matrix

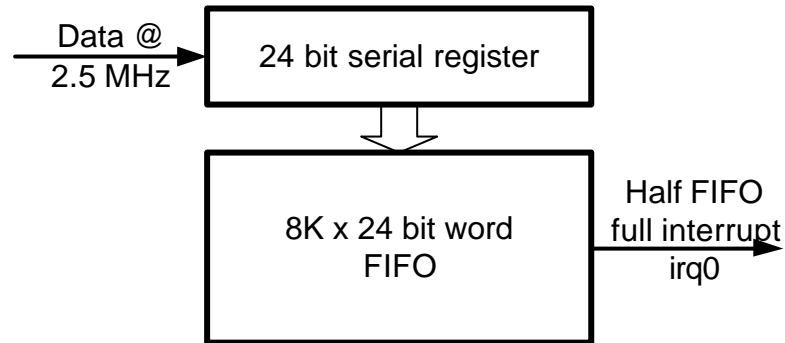
The exec_matrix is actually a two column (A,B) 32bit word matrix. The MS 16 bit of A (AH) defining the function of the full row.



AH		AL	BH	BL
ahh	ahl			
0	0	next int3 period [ms]	Command to subsystem	
1	N=jump [R] to row N	next int3 period [ms]	Initial loop count	Current loop count
2=set mutex	X	next int3 period [ms]	Loop control or no action, depending on ahl	
3=reset mutex	X	next int3 period [ms]	Loop control or no action, depending on ahl	
0		0	Terminate. Disable interrupt int3	

HIFI: the next TC terminate the sequence (disable int3)

High Speed I/F Timing



Typical coadd should last 30 CPU cycles or **1.5 us** (@ 20 MHz)

Operation	Cycles
Read from FIFO	3
Test FIFO empty	3
Mask data word	1
Increment destination address	1
Check for Begin new record	10
Add to dest. address	6
other	6
Total	30

- 24+1 bit data word @ 2.5 MHz => 10 us/word
- 2 simultaneous high speed link transmitting => 5 us/word
- Data formatting must be much shorter than **5 us/word**

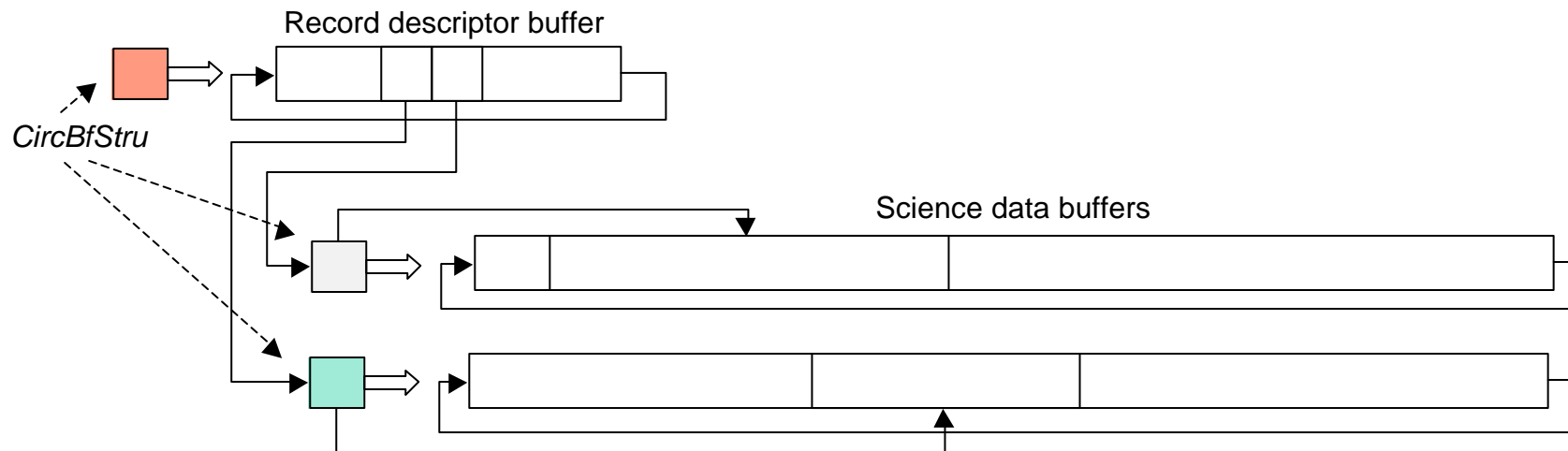
Science data buffer

The record descriptor buffer (RDB) and Science data buffers have an associate CircBufStru variable describing:

```
struct CircBufStru {
    int *pBase;           // pointer to buffer start addr
    int  xIn;             // index to current in (from pBase)
    int  xOut;            // index to current out (from pBase)
    int  len;             // length of circ buffer
}
```

RDB is a vector of elements type RecDesStru describing:

```
struct RecDesStru{
    CircBufStru *pSciCB; // pointer to science data buff associated struct.
    int  sBeg;           // index to start of record (from pSciCB->pBase)
    int  sLen;           // length of the record
    Any other packet/science descriptor
}
```



C Style

- Global variables begin with one **C**apital character
- Startic variables begin with two **C**apital characters
- Pointers are prefixed by “**p_**”
- All global variable are defined in the main module and defined “extern” in the modules where are used
- Each function has a full prototype definition in the module header and defined external (with comment indicating the module source) where it is called
- C++ convention for include files :

```
#ifndef __MY_UNICNAME_H__
#define __MY_UNICNAME_H__
...
#endif // __MY_UNICNAME_H__
```
- One (or more) module(s) for each task
- 4 spaces tabs for indentation
- Version control system using CVS (WinCVS)

C Style

```

/*-----
   $RCSfile: dpuspi.c,v $
   $Revision: 1.10 $
   $Date: 2001/05/28 11:31:02 $
   $Id: dpuspi.c,v 1.10 2001/05/28 11:31:02 cerulli Exp $

Module description:
   ...
   ...
-----*/

//===== Functions in this module =====
void connectToSs(void);           // description

//===== EXTERN FUNCT =====
int incrCircPo(int * p_base, int len, int coll, char * p_buf);

//===== GLOBAL VAR =====
extern struct SsysIO_Param IoPar[]; // description

//===== STATIC VAR =====
...

/*-----
   Funct description: Connect to the right subsystem and send
   (receive) data
-----*/
void connectToSs()
{
  ...
}

```