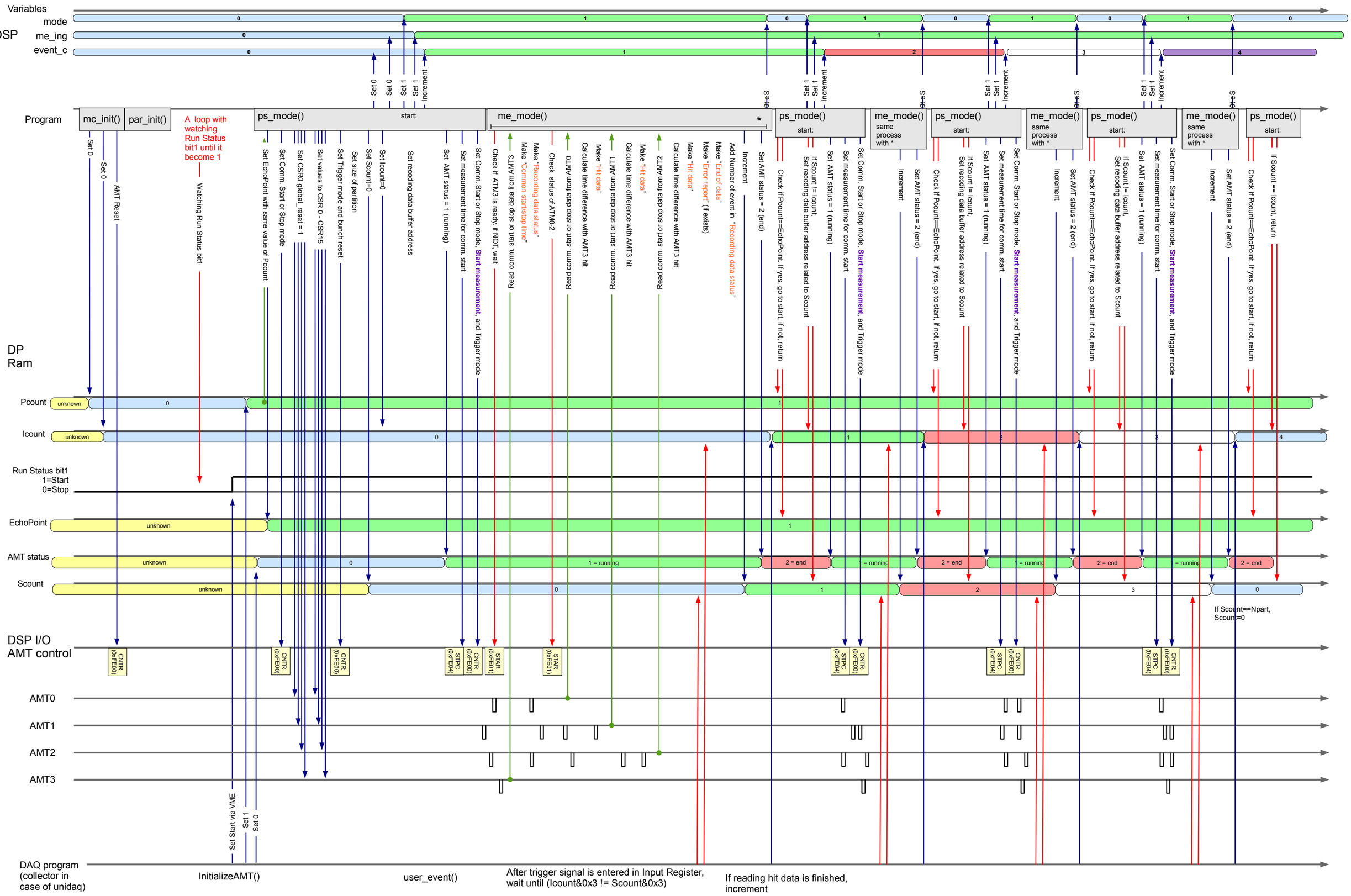
Sequence


Variables
DSP
mode
me_ing
event_c
Program
mc_init()
par_init()
A loop with watching Run Status bit1 until it become 1
ps_mode()
start:
me_mode()
*
me_mode()
same process with *
Set 0
Set 1
Increment
Set EchoPoint with same value of Pcount
Set Comm. Start or Stop mode
Set CSR0 global_reset = 1
Set values to CSR 0 - CSR15
Set size of partition
Set Trigger mode and bunch reset
Set Scount=0
Set Icount=0
Set recoding data buffer address
Set AMT status = 1 (running)
Set measurement time for comm. start
Set Comm. Start or Stop mode, Start measurement, and Trigger mode
Check if ATM3 is ready, if NOT, wait
Read comm. start or stop data from AMT3
Make "Common start/stop time"
Make "Recording data status"
Check status of ATM0-2
Read comm. start or stop data from AMT0
Calculate time difference with AMT3 hit
Make "Hit data"
Read comm. start or stop data from AMT1
Calculate time difference with AMT3 hit
Make "Hit data"
Read comm. start or stop data from AMT2
Calculate time difference with AMT3 hit
Make "Hit data"
Make "End of data"
Make "Error report" (if exists)
Add Number of event in "Recording data status"
Increment
Set AMT status = 2 (end)
Check if Pcount==EchoPoint. If yes, go to start, if not, return
If Scount != Icount, Set recoding data buffer address related to Scount
Watching Run Status bit1
AMT Reset
DP Ram
Pcount
Icount
unknown
Run Status bit1
1=Start
0=Stop
EchoPoint
AMT status
1 = running
2 = end
Scount
If Scount==Npart, Scount=0
If Scount == Icount, return
DSP I/O
AMT control
CNTR (0xFE00)
STPC (0xFE04)
STAR (0xFE01)
AMT0
AMT1
AMT2
AMT3
Set Start via VME
DAQ program (collector in case of unidaq)
InitializeAMT()
user_event()
After trigger signal is entered in Input Register, wait until (Icount&0x3 != Scount&0x3)
If reading hit data is finished, increment