


UNILAC timing essentials

- UNILAC has to be operated at 50 Hz without any interruptions in any of the following (not exhaustive)
 - (ex)change of schedule/pattern
 - change of chain events, e.g. time of event or addition/removal of events ⇒ switch between static event sets
 - data supply to devices ⇒ single device: continue operation; large sections: cont. op. w/o beam
- Synchronised wrt. power grid 50 Hz
- Online actions (not exhaustive):
 - Profile grid guard: reduce beam pulse length and execution rate with beam for all chains affected on operators request
 - Change beam pulse length on operators request
 - Interlock: execute affected chains w/o beam on request

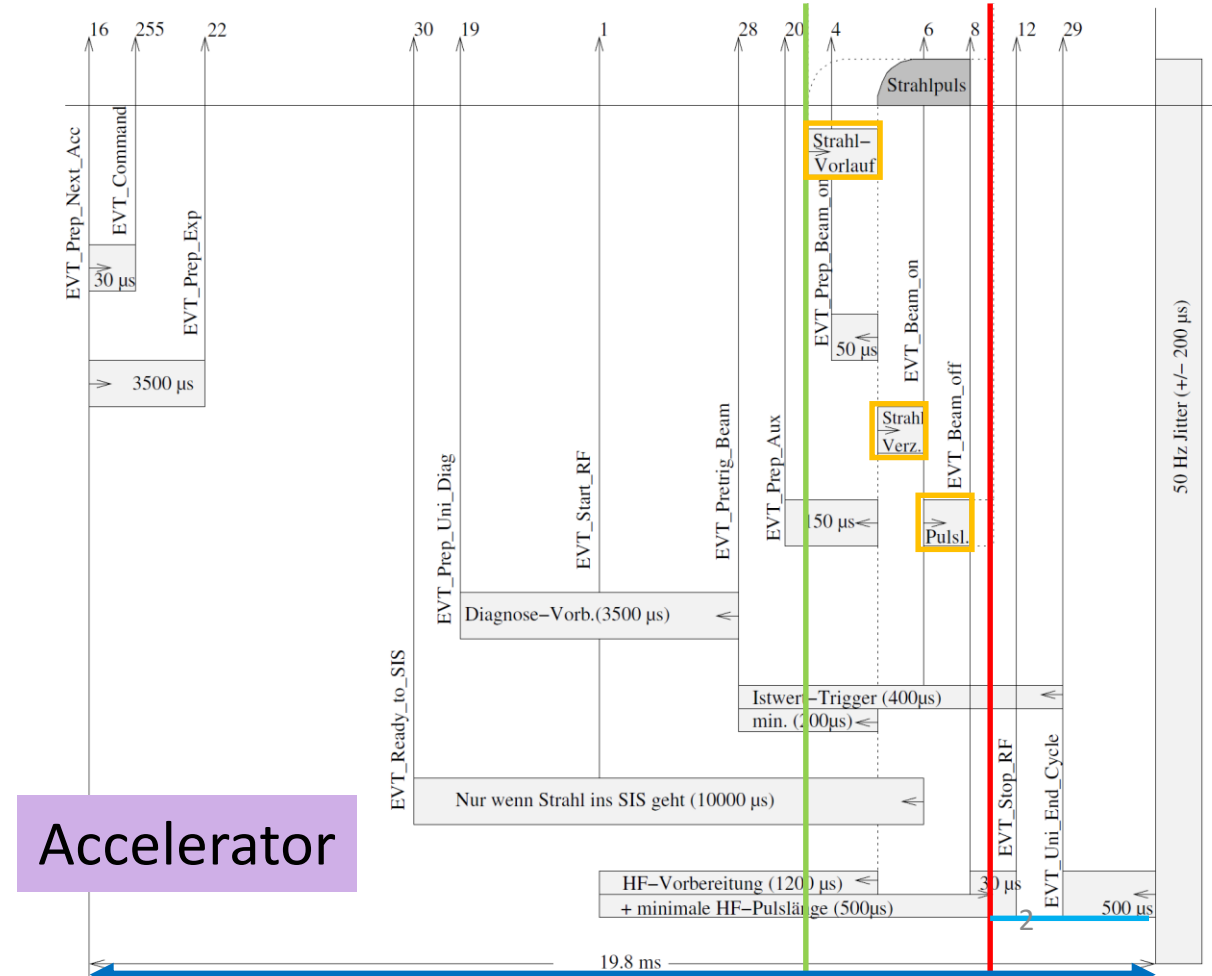
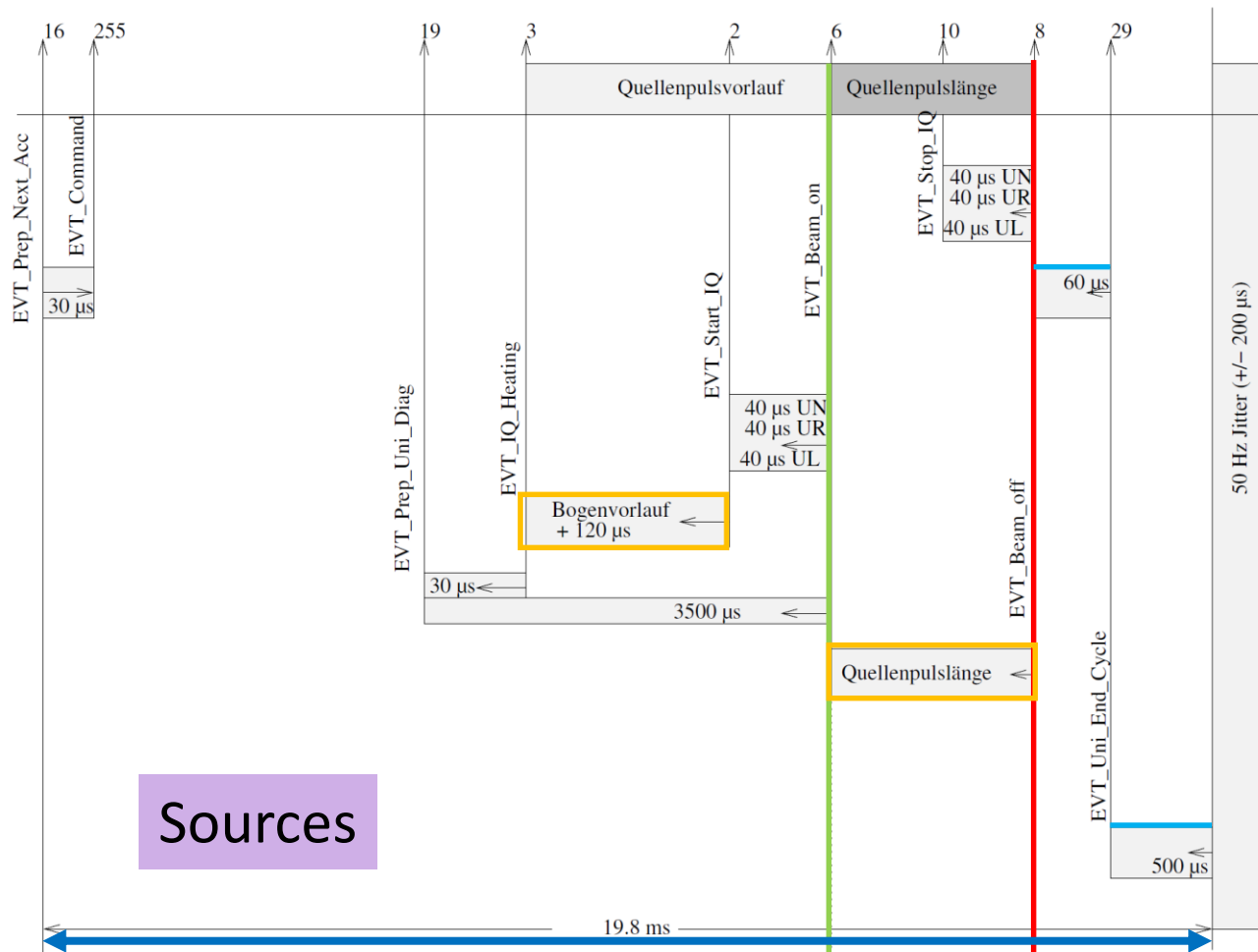
2 principle UNILAC event sequences: Fix – except pulse lengths

 Operating parameters (pulse lengths)

 Cycle length (20 ms)

 Connection source-accelerator timing
(ion beam pulse start & stop at source)

 Connection to absolute time



New UNILAC timing basics (draft proposal)

- Preplanned schedules:
 - based on SIS18 beam requirements and source availabilities \Rightarrow length equals SIS18/FAIR-pattern (~1s to 100s of seconds?), UNILAC may require certain length of SIS18/FAIR-pattern
 - filled in with UNILAC users, UNILAC machine needs added
 - run periodically by DM, $50 \cdot n$ chains per second
 - one schedule per SIS18/FAIR-pattern, extra schedule for rarely executed FAIR chains, plus UNILAC standalone schedule, and others?
 - switch between schedules at starting point without interruption of 50 Hz operation
 - contains chain indices (pointers, wrappers), hundreds of executions of several chains in one schedule
- Standalone chains:
 - single chain identity may be executed multiple times in one schedule and used in several schedules
 - set of at least 16 chains available in timing system, any subset may be used in any schedule
 - chains may be executed simultaneously in arbitrary combinations (only one chain per timing group)
 - changes applied to a chain have to take effect for all future executions at one point in time
 - persistent identification chain \Leftrightarrow virtual accelerator, over different schedules
 - one single UNILAC chain may be combined (“gekoppelt”) with several SIS18 chains
- Event sequence, execution:
 - All UNILAC chains share similar, short event sequences of fixed length; two principle event sequences: 1) sources & LEPT; 2) all other timing groups after chopper
 - only two or three event sequences needed per chain
 - beam processes and events of one chain run in parallel in all timing groups \Rightarrow bursts of events at specific times
 - changes to the event sequence of one chain have to take effect in all timing groups simultaneously
 - generate several versions of event sequence in advance or on-the-fly: standard, shortened beam pulse, no beam pulse
 - change beam pulse length: generate new standard event sequence, send to DM, reset pointer/wrapper
- Timing groups:
 - ~45 timing groups (~14 would be sufficient)
 - all timing groups of one chain share the same event sequences (except source & LEPT and end of TK)
 - provide MIL-timing derived from adequate timing group

Example schedule

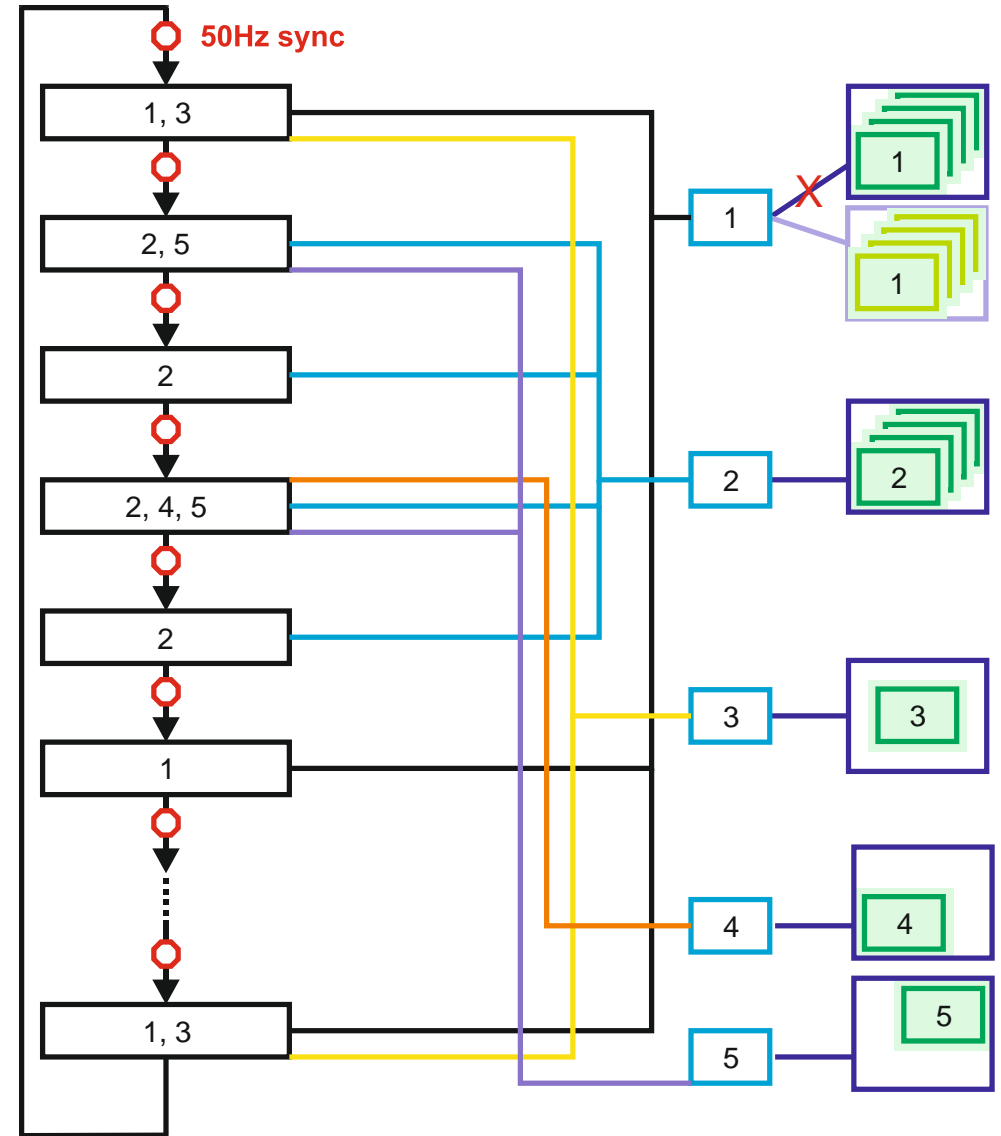
BPC-Sequences							
Cycle	=VirtAccs	HSI	HLI	ALV	EH	TK	SIS18
1	1 ^A +3	HSI01	HLI03	ALV01	-	TK01	SIS_BPC_A
2	2+5	-	HLI02	ALV02	EH02	TK05	
3	1+6	HSI01	HLI06	ALV01	-	TK01	
4	2+4	HSI04	HLI02	ALV02	EH02	-	
5	2+5	-	HLI02	ALV02	EH02	TK05	
6	1 ^B +3	HSI01	HLI03	ALV01	-	TK01	SIS_BPC_B
...	

Schedule

FAIR-Pattern

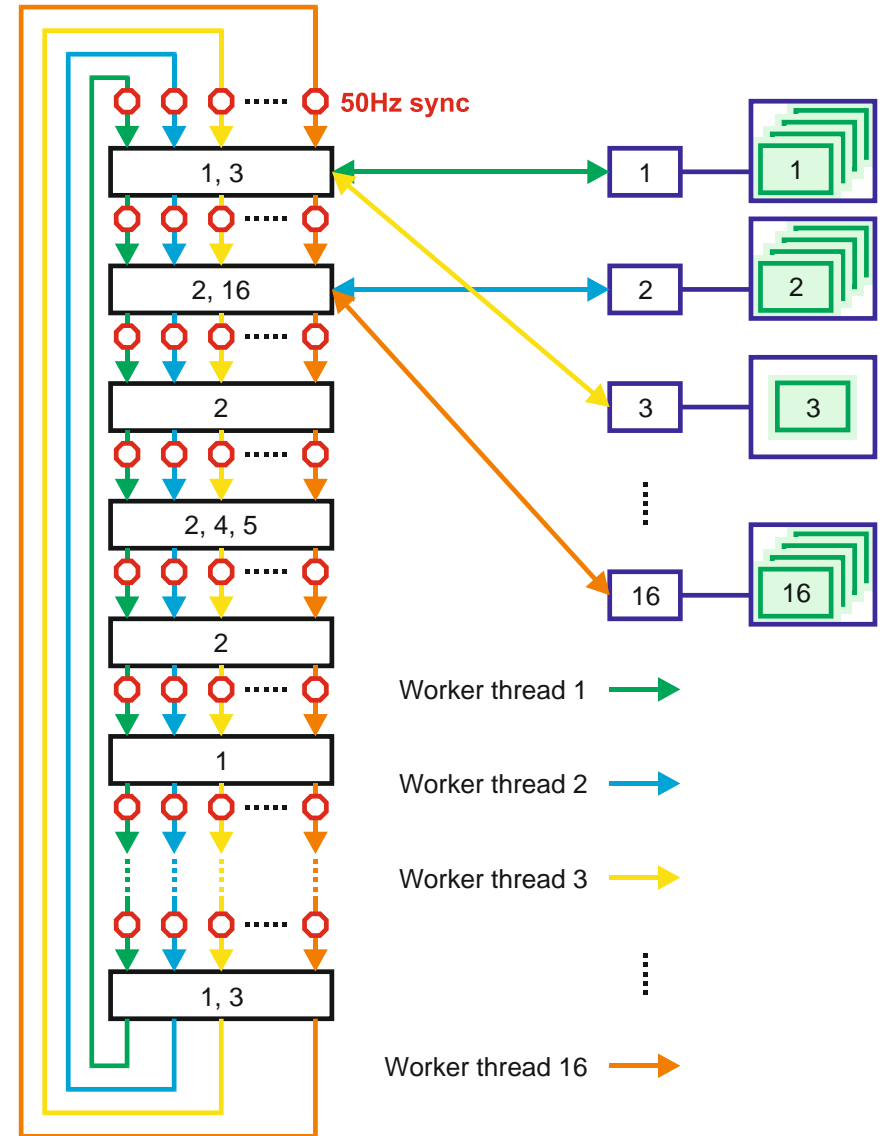
Merge beam processes, insert abstraction layer

- Merge beam process nodes to one BPC node
 - Contains all timing messages, hence all information for one beam production chain
- Hide BPC nodes by static wrappers
 - Contain only pointer to BPC node
 - Many pointers in supercycle direct to static wrapper \Rightarrow pointers and supercycle not affected by BPC exchange
- Supercycle and wrappers contain only BPC index
- Exchange of beam process
 - Generate **one** new node with new version of BPC timing messages
 - Change **one** pointer from wrapper to this node



Supercycle execution

- Single supercycle
- Multithreaded, one worker thread per BPC
 - Branch to BPC node if index in cycle
 - Generate timing messages for BPC
 - Callback
 - Go to next supercycle node
- Synchronization mechanism available
- Arbitration mechanism available
 - thread with nearest event gets executed next



Timing Groups: EH dipole (and quads)

Single power supply serves magnets in different timing groups

