

Baiyang: High Performance Open-Source DDR Memory Controller IP

The XiangShan Team
HPCA 26 @ Sydney, Australia
January 31, 2026

Contents

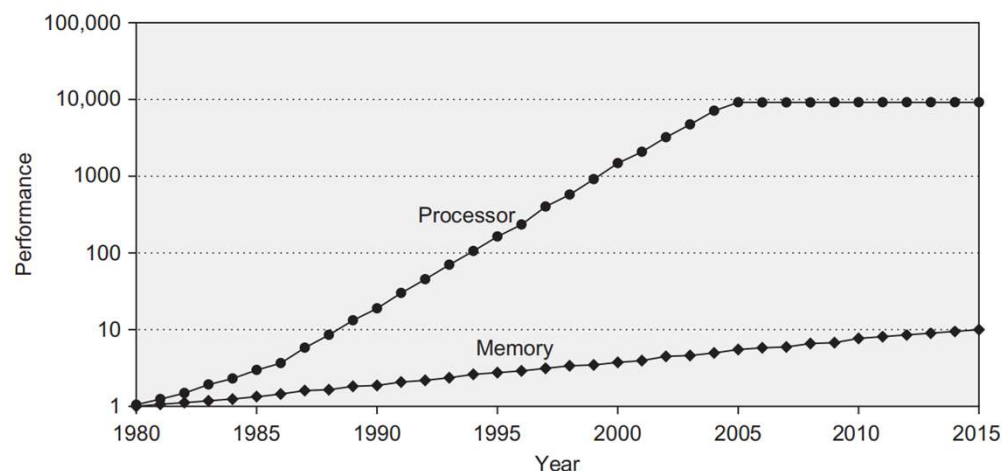
- Background & Motivation
- Baiyang: High-Performance Memory Controller IP
- Future Perspectives

Contents

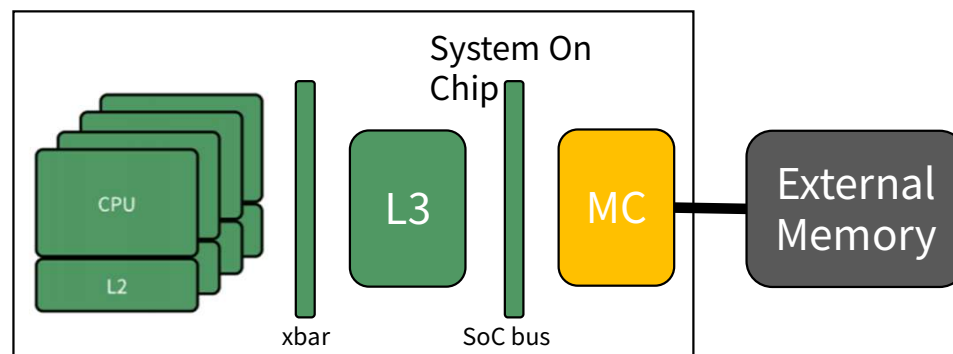
- **Background & Motivation**
- Baiyang: High-Performance Memory Controller IP
- Future Perspectives

Why Memory Controller (MC)?

- **Memory wall issue**
 - Memory performance development: much slower than processor
 - Memory performance is a **primary factor in system performance**
- **High performance MC IP**
 - XiangShan processor development
 - Core-L1-L2-L3 Cache is evolving rapidly
 - Open-Source MC is still dumb
 - **Cannot meet demand of processor**
 - Lack of joint optimization and innovation



The Performance Gap between Processor and Memory, Starting with 1980 performance as a baseline [1]



[1] HENNESSY J L, PATTERSON D A. Computer architecture: A quantitative approach[M]. 6th ed: Page 80, Figure2.2.

Current MC IP

- ❶ Commercial MC: licenses are monopolized, hindering independent development
- Expensive: **one-time DDR5 license fee near 1 Million\$**
- Hundreds of configurable parameters: **difficult to explore optimization space**



- ❷ Open-source High-performance MC IP: limited function and performance, not tape-out ready
- LiteDRAM^[1]: only support 1250MT/s, DDR3
- OPRECOMP^[2]: only support single burst request, DDR4-1600
- OpenDRAM^[3]: binding with Xilinx PHY

[1] <https://github.com/enjoy-digital/litedram>

[2] <https://github.com/oprecomp/oprecomp>

[3] <https://github.com/FanosResearch/OpenDRAM>

Contents

- Background & Motivation
- **Baiyang: High-Performance Memory Controller IP**
- Future Perspectives

Baiyang MC IP: design overview

• Design requirements and considerations

- Adaptive to different memory scenarios

① Multi-frequency support, DDR4-1600/2400/3200

parameterized &
configurable MC

② Compatibility with various modules and DRAM chips

③ Support for multiple interface/pin configurations

modular decomposition
and reuse

④ Optimized background operations

Advanced features :
dynamic refresh, schedule,
cache and prefetch.

⑤ Basic read/write functions & advanced features

⑥ Compatibility with PHYs from multiple vendors

supports bus protocols such
as DFlx and AXI4.

⑦ Support for multiple high-speed interconnect buses

Baiyang architecture

- **Basic MC architecture with advanced features and multiple interface support**

Parameter configuration

- Supports different DDR modules and frequencies

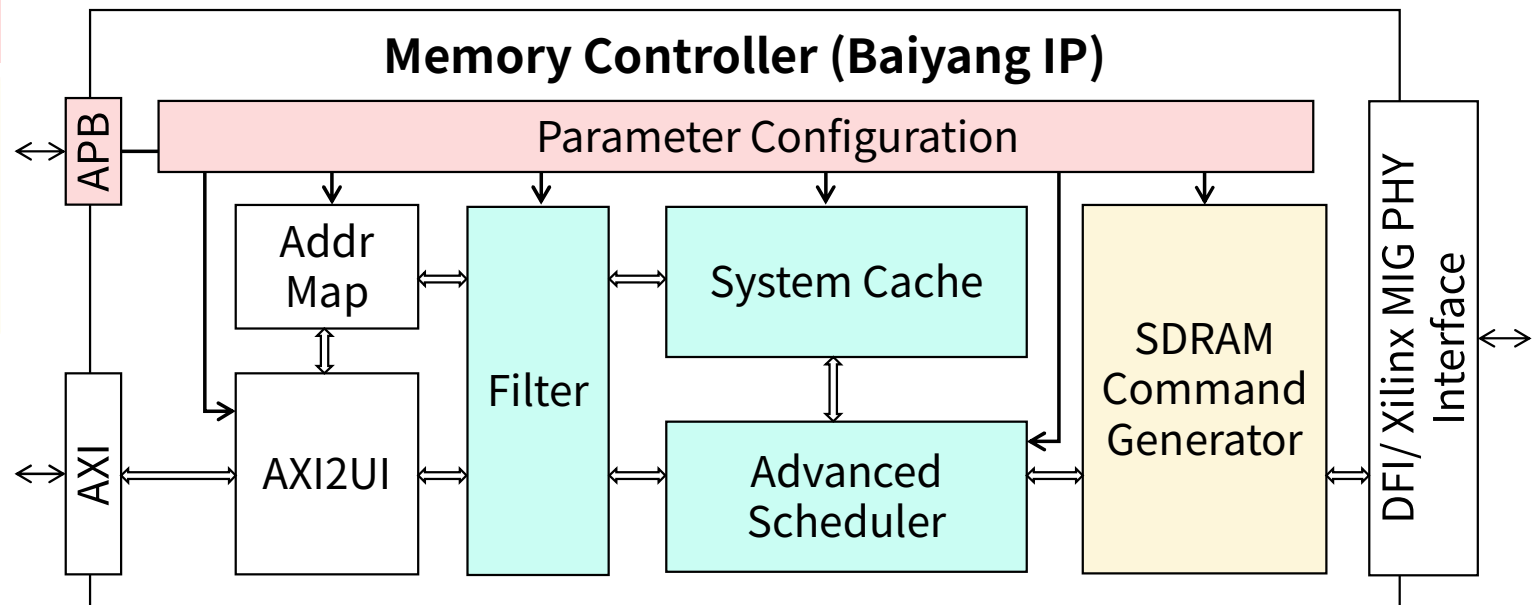
Controller Modules

- Decoupling DDR timing parameters from controller functionality

Rich functionality

- Advanced and configurable features: dynamic refresh, scheduling, cache, prefetch

<https://github.com/OpenXiangShan/YuQuan>




Functional and performance testing

- Functionality: integrated with Xiangshan Kunminghu core in Palladium verification environment, successfully boots Linux

- Performance: integrated Xiangshan Kunminghu core achieves SPEC CPU2006 score 14/GHz, **approaching the performance level of commercial MC IP**

```
[ 0.000094] pid_max: default: 4096 minimum: 301
[ 0.000125] Mount-cache hash table entries: 512 (order: 0, 4096 bytes, linear)
[ 0.000143] Mountpoint-cache hash table entries: 512 (order: 0, 4096 bytes, linear)
[ 0.000235] ASID allocator using 16 bits (65536 entries)
[ 0.000269] clocksource: jiffies: mask: 0xffffffff max_cycles: 0xffffffff, max_idle_ns: 191126044627500
[ 0.000368] clocksource: Switched to clocksource riscv_clocksource
[ 0.000618] workingset: timestamp_bits=62 max_order=13 bucket_order=0
[ 0.001446] Freeing unused kernel image (initmem) memory: 520K
[ 0.001461] Kernel memory protection not selected by kernel config.
[ 0.001475] Run /init as init process
```



```
Hello, XiangShanCore 0: HIT GOOD TRAP at pc = 0x1063c
Core-0 instrCnt = 21515634, cycleCnt = 19493019, IPC = 1.103761
!
DIFFTEST WORKLOAD DONE at cycle          19495591
--- HW execs 8360 tbcall syncs
--- HW execs 8376 memory read 8376 write
--- HW execs 39018127 evals 171 bevals (0 bp) 42634625 cfclks 8362 tbsyncs 51200833 fclks in 36.11 sec (0.77%)
--- HW execs emulator busy 29.16 sec (80.77%)
--- HW execs 50948420 gfifo-wait evals (46200785 gbfull, 0 lbfull, 4747635 gftbsync) in 17.42 sec (48.26%)
--- HW execs 8360 output events
--- HW execs emulator command line session time 40.67 sec (36.52 CPU sec)
```

	commercial IP	baiyang_V0.92	Perf diff
SPEC Int			
libquantum	33.778	33.371	-1.20%
mcf	19.991	19.64	-1.76%
gcc	15.395	15.233	-1.05%
omnetpp	13.447	12.581	-6.44%
xalancbmk	23.531	23.317	-0.91%
sjeng	10.213	9.848	-3.57%
perlbench	11.92	11.781	-1.17%
gobmk	10.053	9.946	-1.06%
hmmer	13.851	13.847	-0.03%
bzip2	8.375	8.368	-0.08%
astar	9.691	9.602	-0.92%
h264ref	18.834	18.71	-0.66%
all	14.48387247	14.25419253	-1.59%

Contents

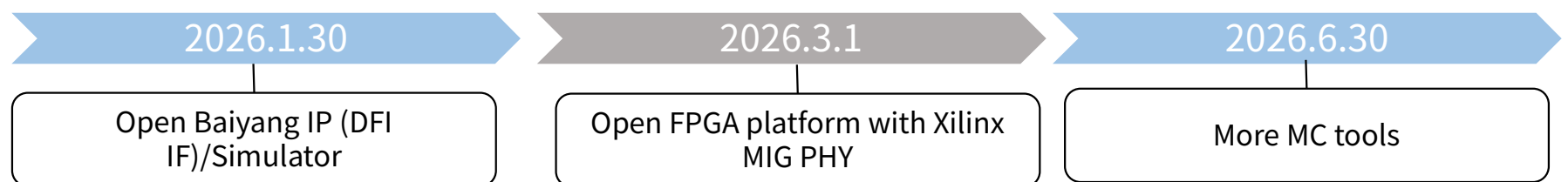
- Background & Motivation
- Baiyang: High-Performance Memory Controller IP
- **Future Perspectives**

Future Perspectives

- Taped out on a 55nm node
 - Compatible with DDR3
 - Compliant with DFI 2.1 protocol
- Evolved for DDR5, DFI 5.1
- Agile development and verification tools
 - Trace-based memory controller simulation and FPGA verification
 - Scaling on multi-core CPU system

<https://github.com/OpenXiangShan/YuQuan>

Open-source
RoadMap





Thanks for listening!

More info:

<https://github.com/OpenXiangShan/YuQuan>