Design of the frontend for LEN5, a RISC-V Out-of-Order processor

Candidate:
Marco Andorno

Supervisor:
prof. Maurizio Martina

POLITECNICO
DI TORINO

Master's thesis in Electronic Engineering

Academic year 2018-2019

LEN5 overview

RISC-V

- Open source ISA, which allows for open source hardware.

RISC-V

- Open source ISA, which allows for open source hardware.
- Modular ISA, that provides extensions to tailor the architecture to the design needs.

Speculative out-of-order core

Used in every modern high-performance processor, because it allows the best exploitation of ILP.

Speculative out-of-order core

Used in every modern high-performance processor, because it allows the best exploitation of ILP.

Based on three pillars:

1. Dynamic pipeline scheduling

Speculative out-of-order core

Used in every modern high-performance processor, because it allows the best exploitation of ILP.

Based on three pillars:

1. Dynamic pipeline scheduling
2. Branch prediction

Speculative out-of-order core

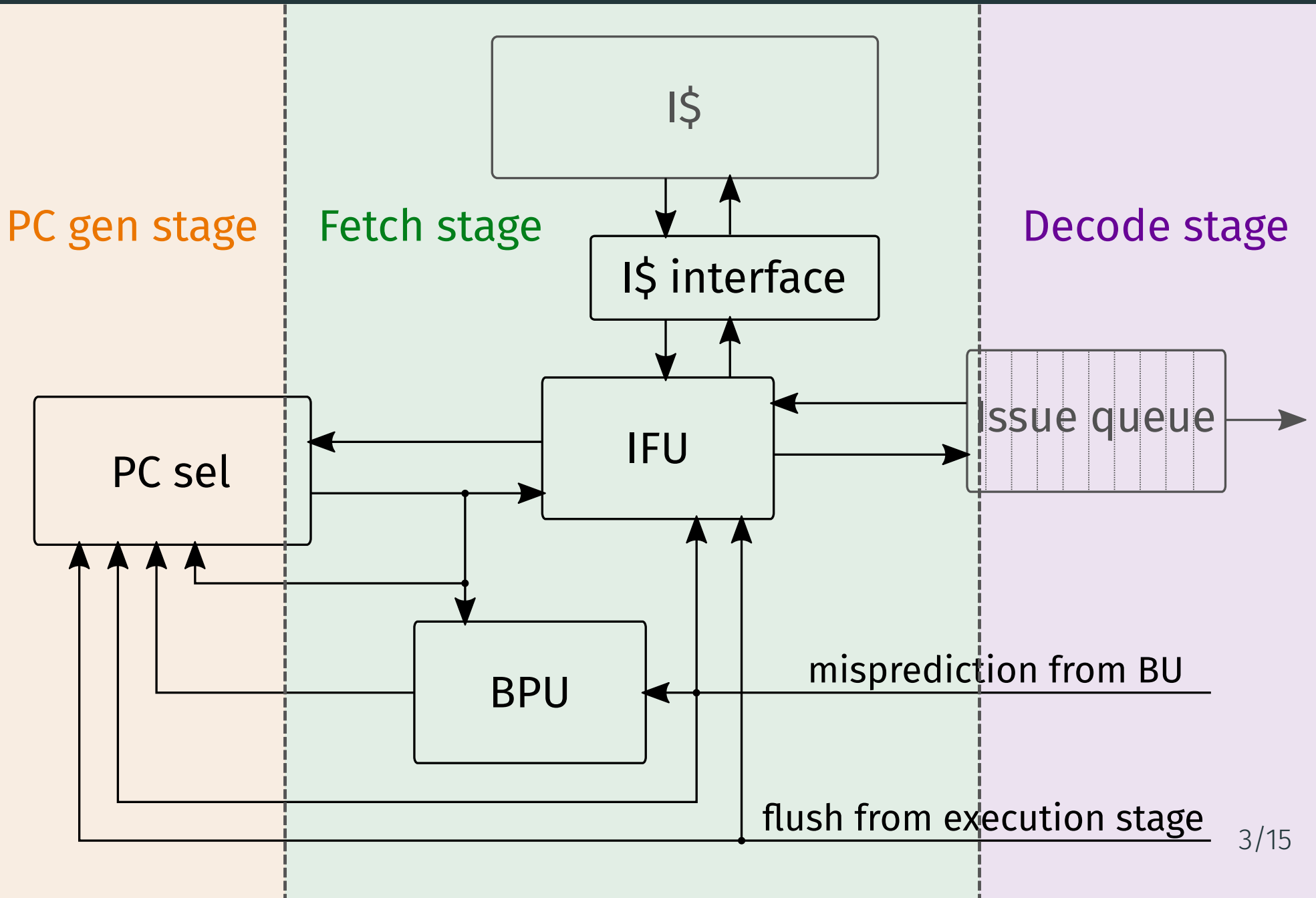
Used in every modern high-performance processor, because it allows the best exploitation of ILP.

Based on three pillars:

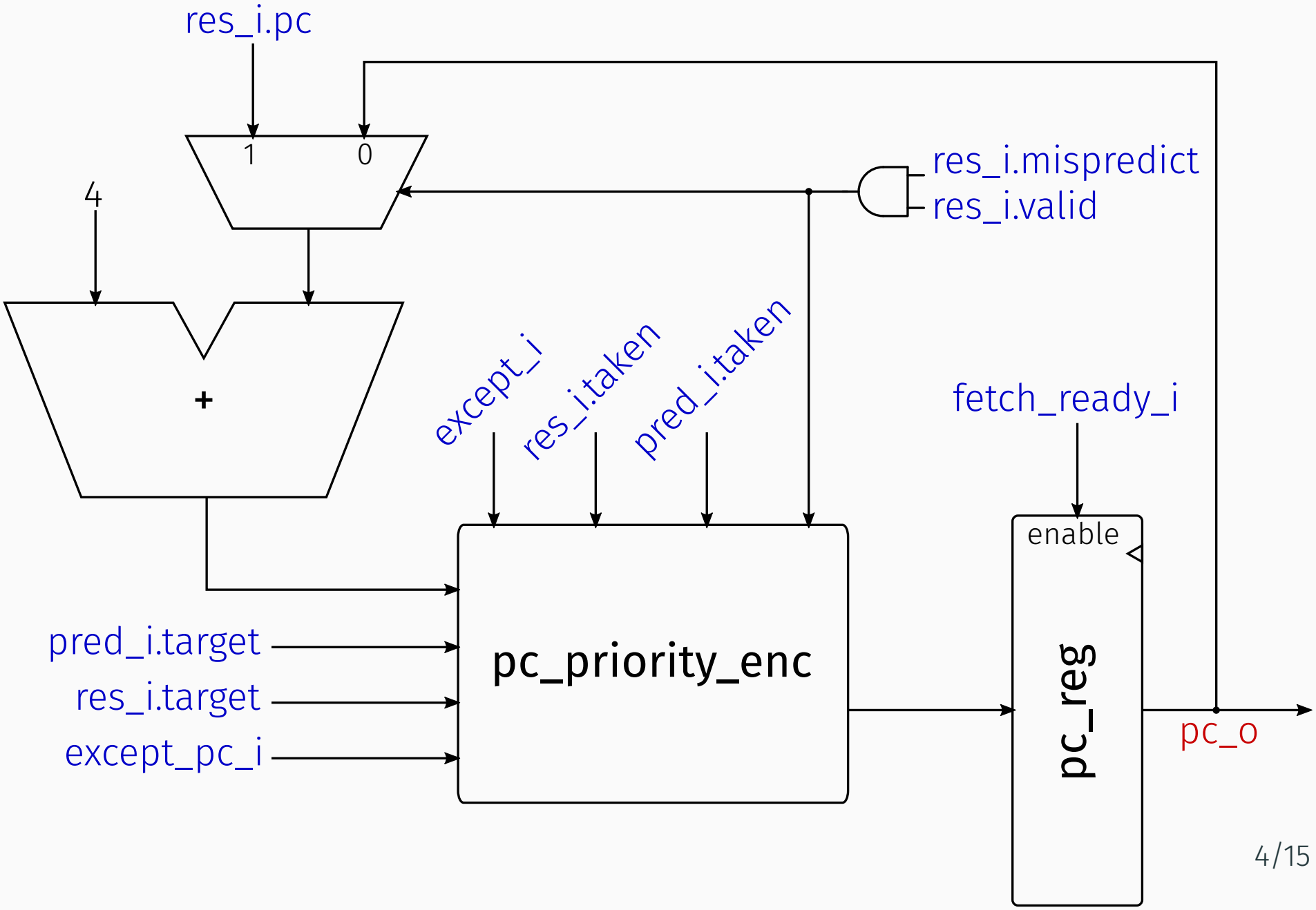
1. Dynamic pipeline scheduling
2. Branch prediction
3. Speculative execution

Frontend design

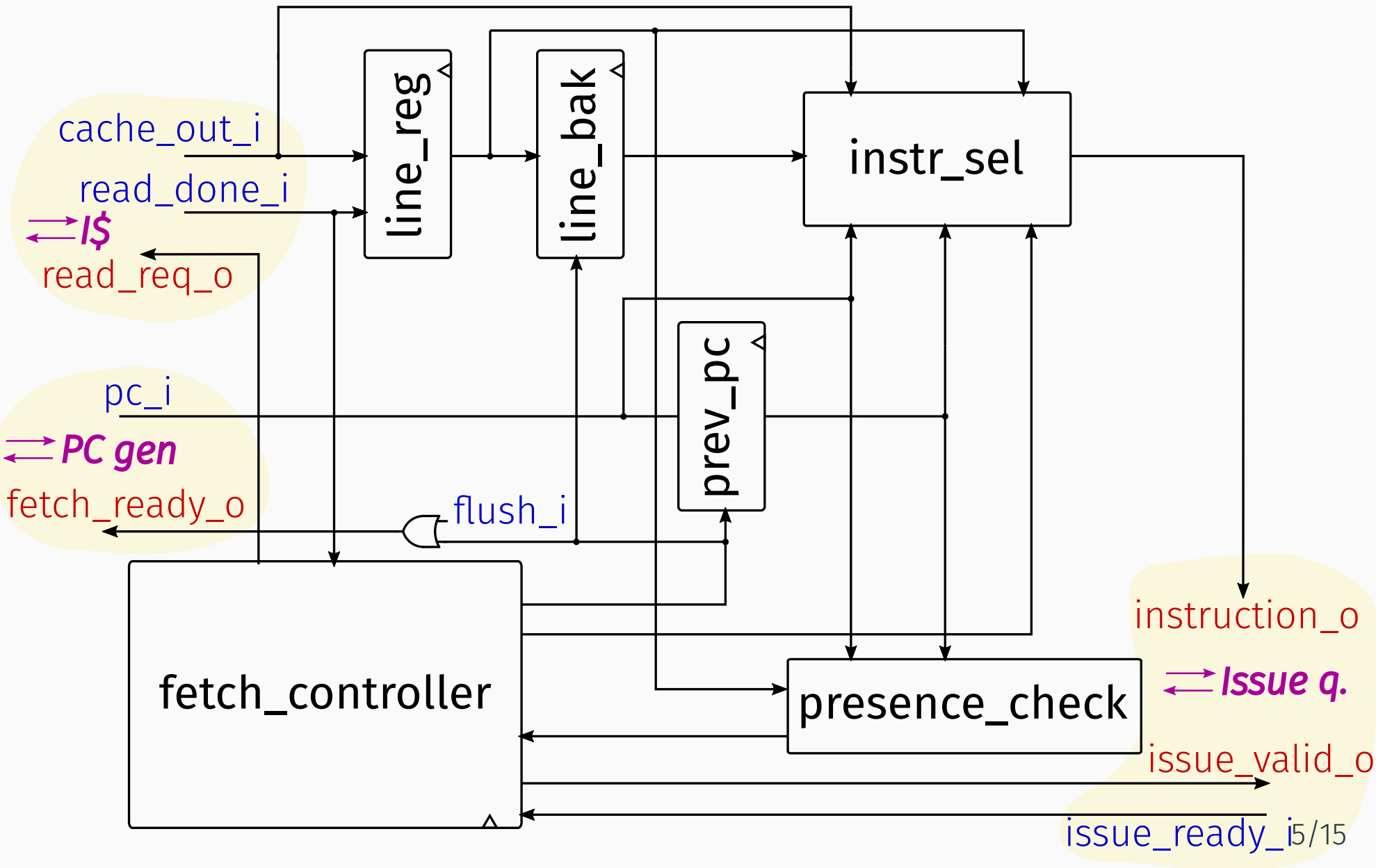
Frontend overview



PC gen stage

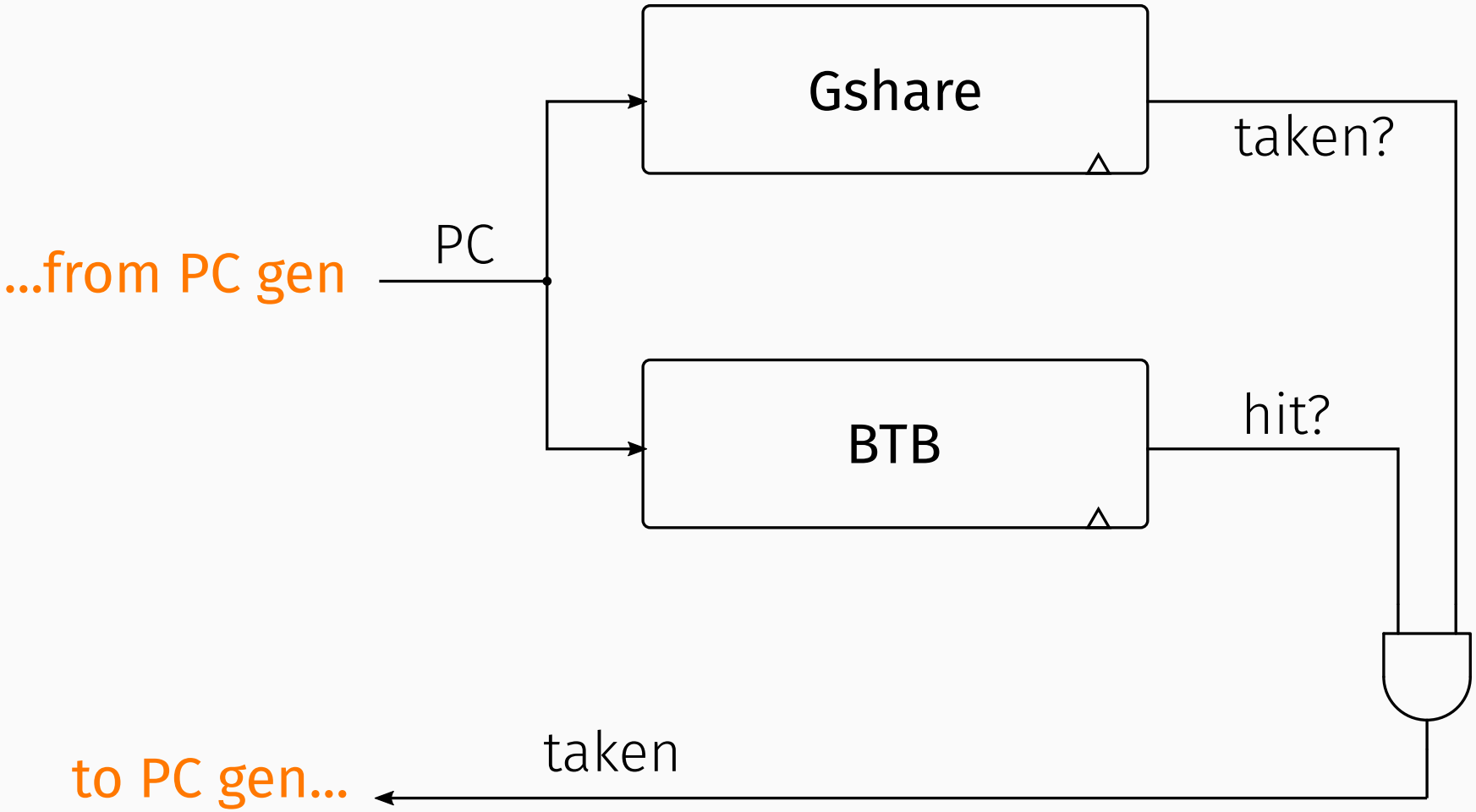

res_i.pc
1
0
4
+
res_i.mispredict
res_i.valid
except_i
res_i.taken
pred_i.taken
fetch_ready_i
enable
pred_i.target
res_i.target
except_pc_i
pc_priority_enc
pc_reg
pc_o

Instruction Fetch Unit (IFU)

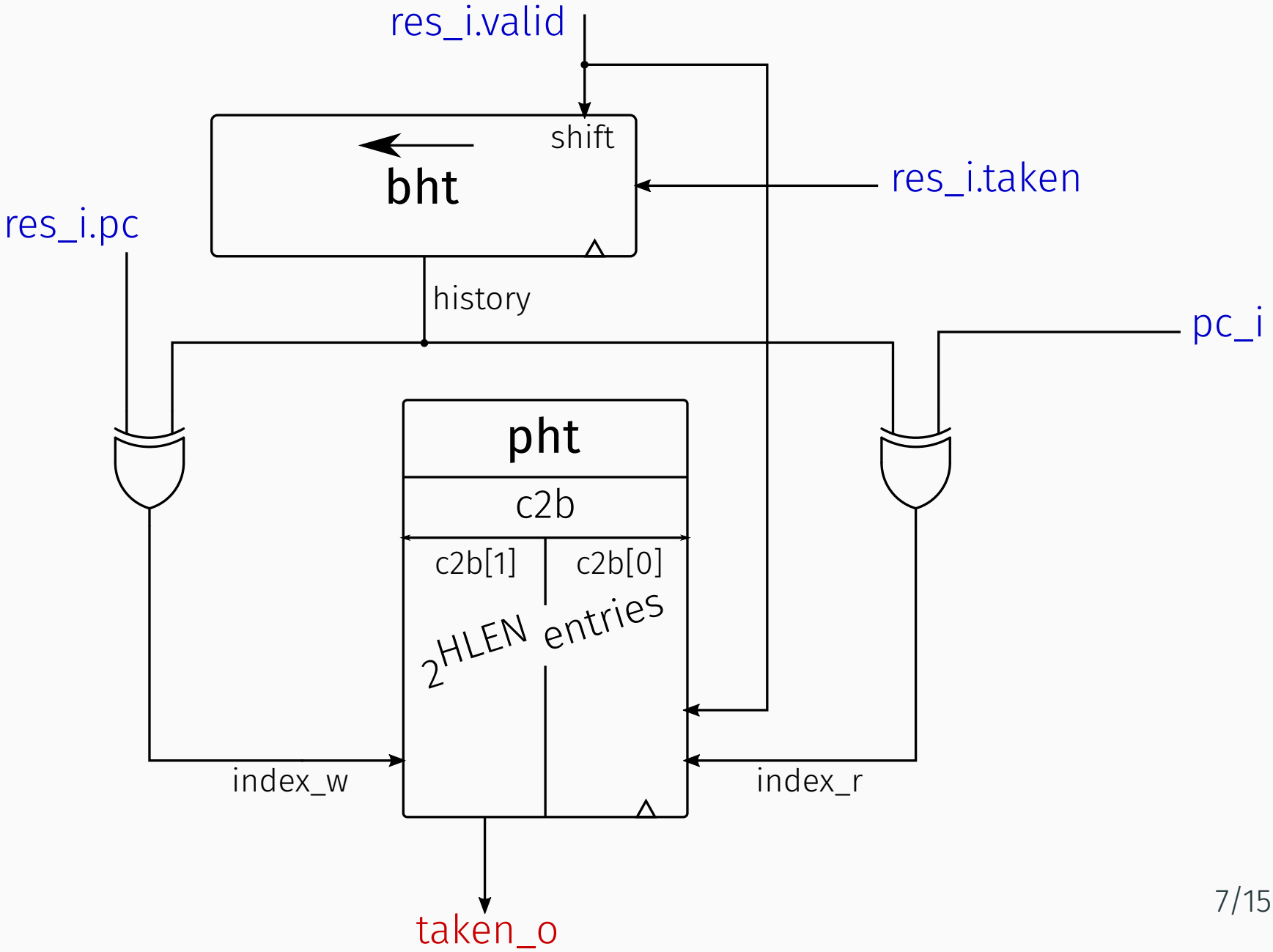


Branch management

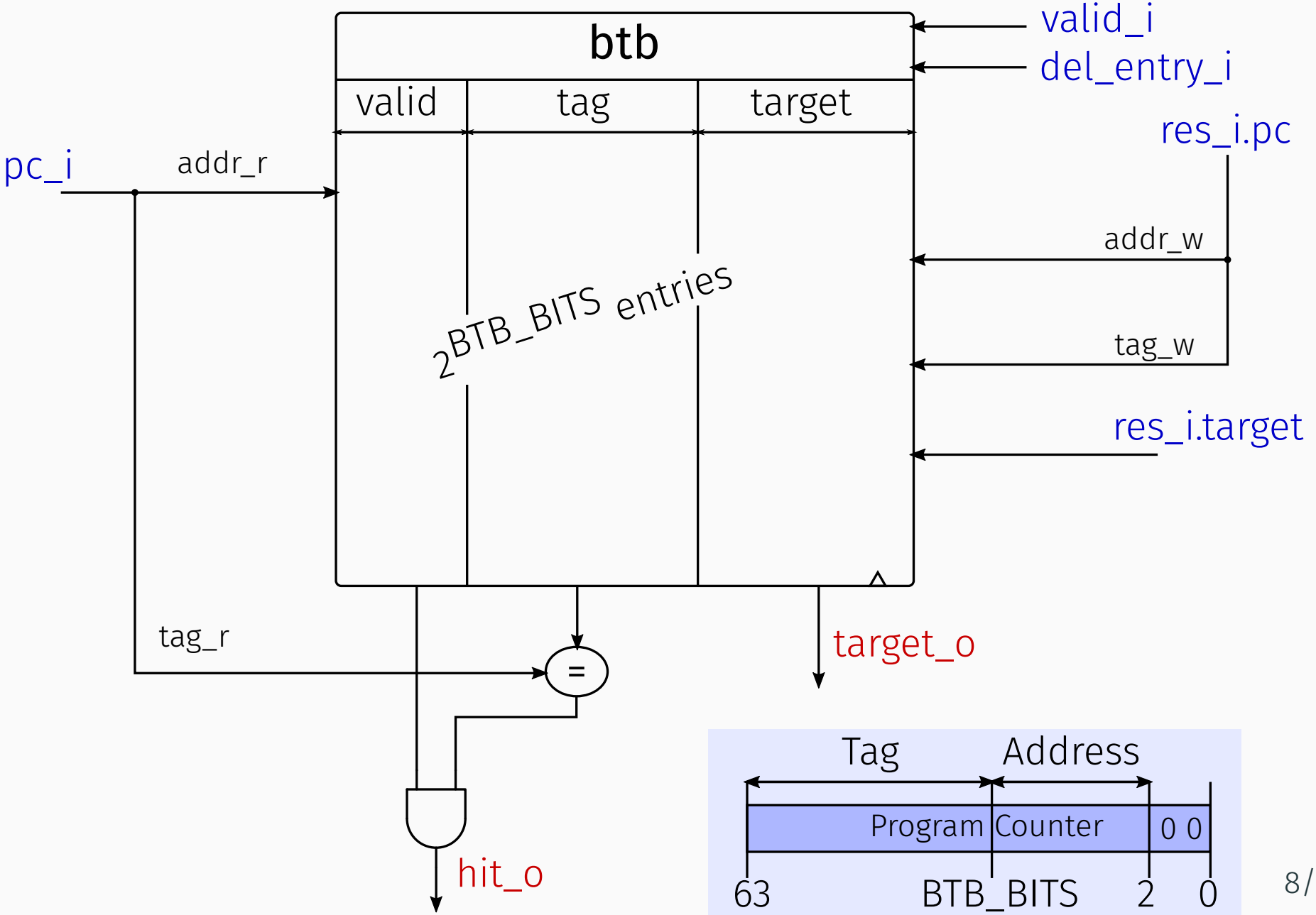
Branch Prediction Unit (BPU)



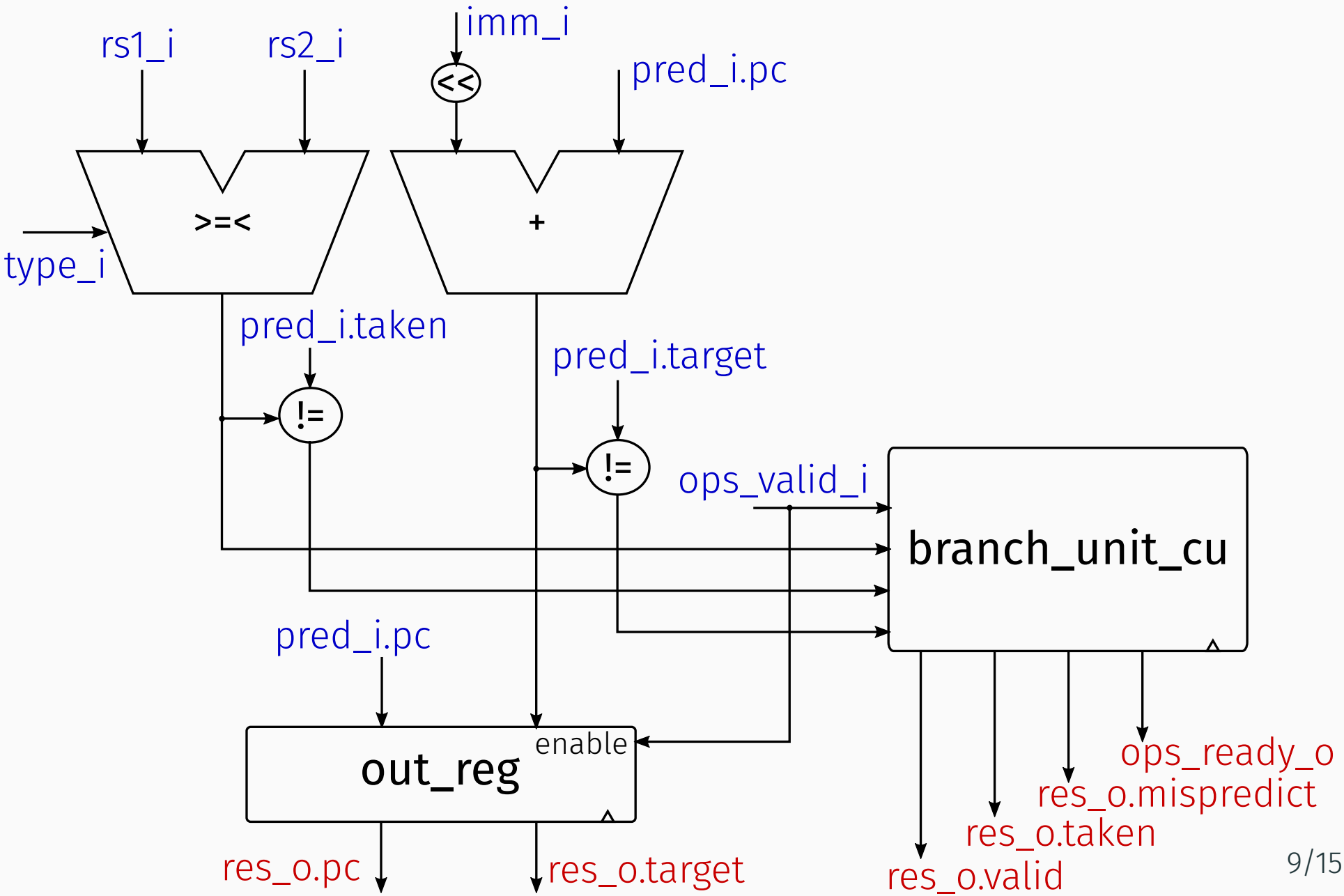
Gshare branch predictor



Branch Target Buffer (BTB)



Branch unit

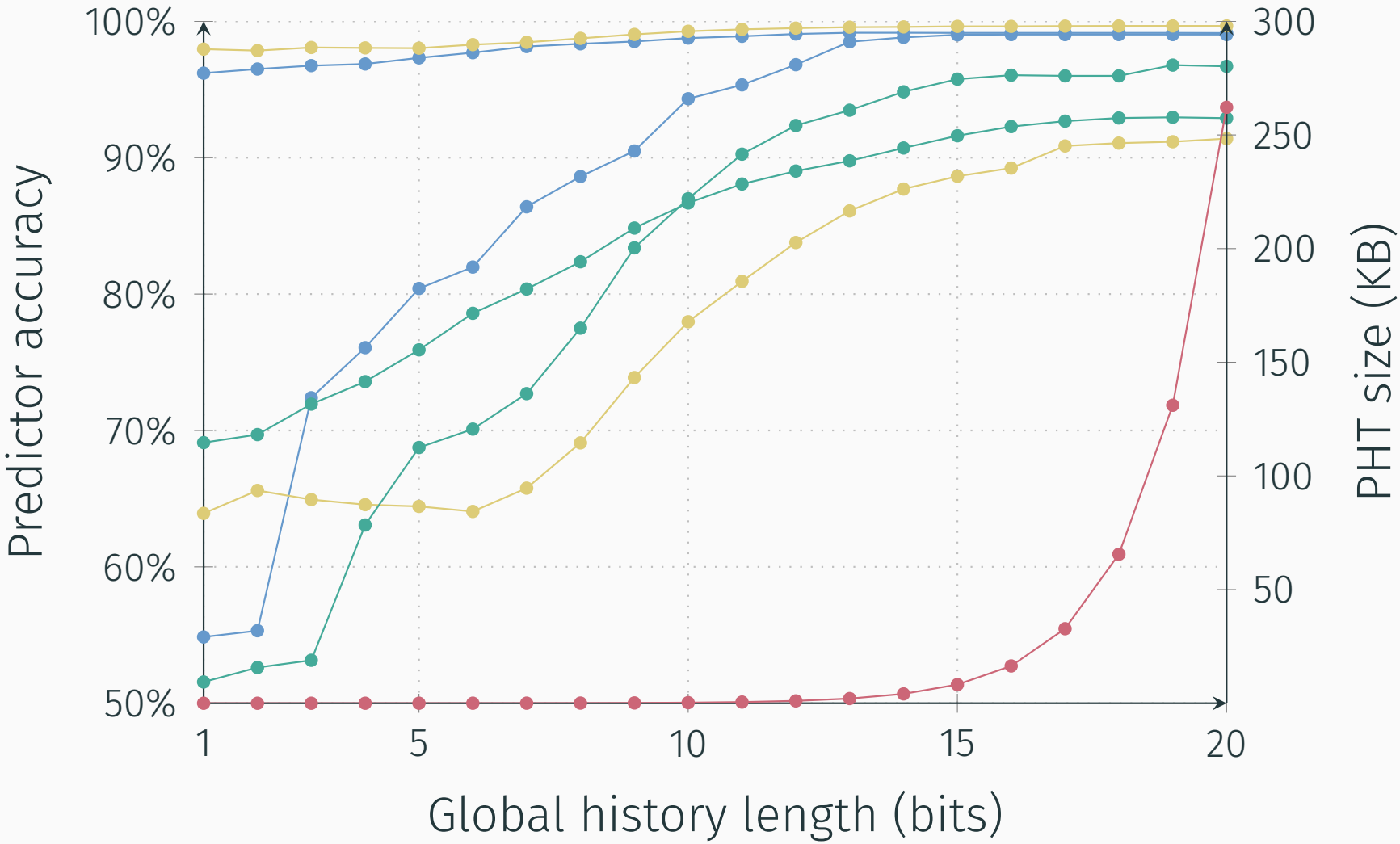

rs1_i
rs2_i
imm_i
<<
pred_i.pc
>=<
+
type_i
pred_i.taken
pred_i.target
!=
!=
ops_valid_i
branch_unit_cu
pred_i.pc
enable
out_reg
ops_ready_o
res_o.mispredict
res_o.taken
res_o.valid
res_o.pc
res_o.target

BPU update actions

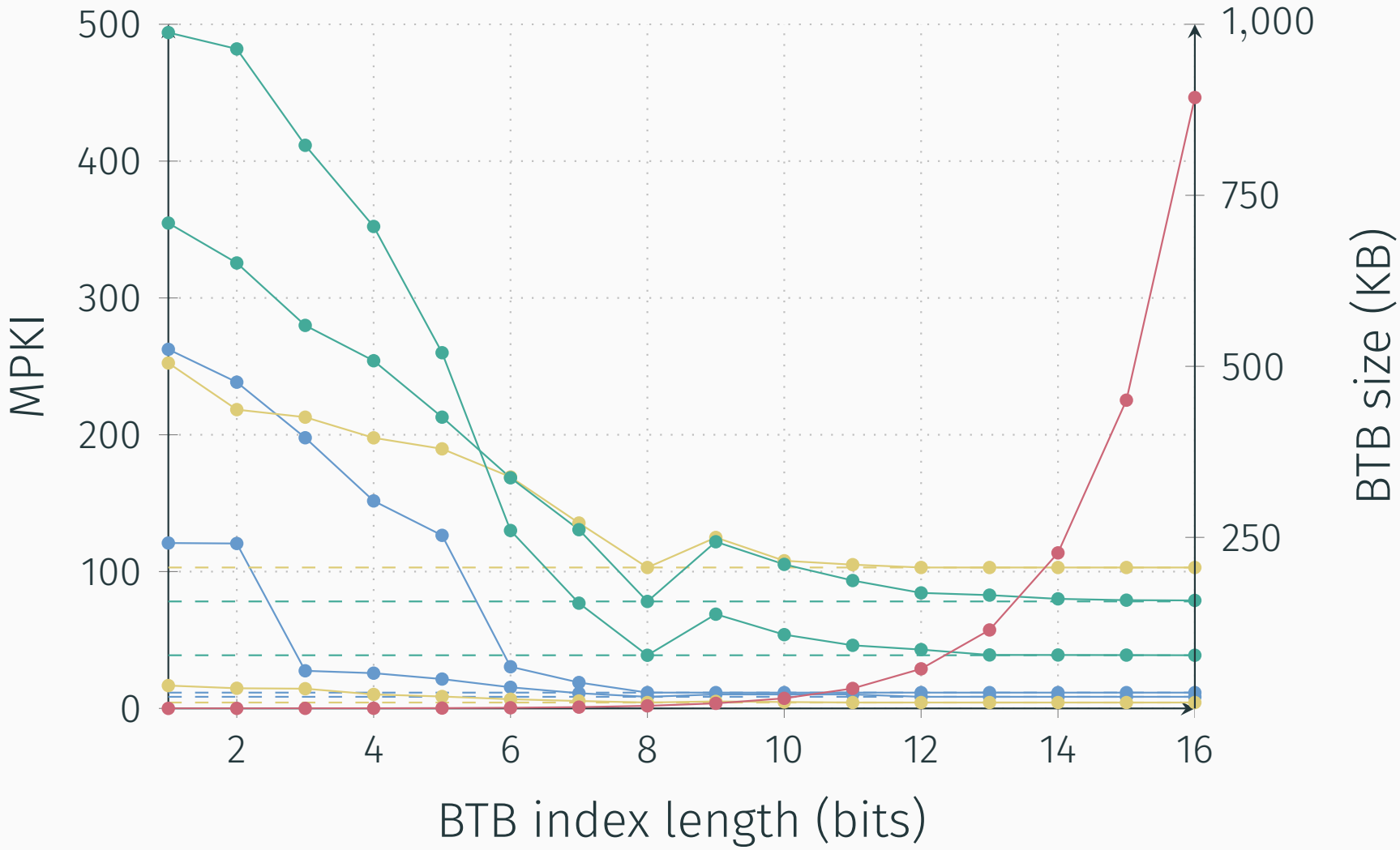
Prediction	Resolution	Target	Action
Taken	Taken	✔	Increment 2-bit counter
		✖	Increment 2-bit counter Update BTB entry Flush, go to right target
	Not taken	–	Decrement 2-bit counter Remove BTB entry Flush, go to branch PC+4
Not taken	Not taken	–	Decrement 2-bit counter
	Taken	–	Increment 2-bit counter Add BTB entry Flush, go to right target

Results

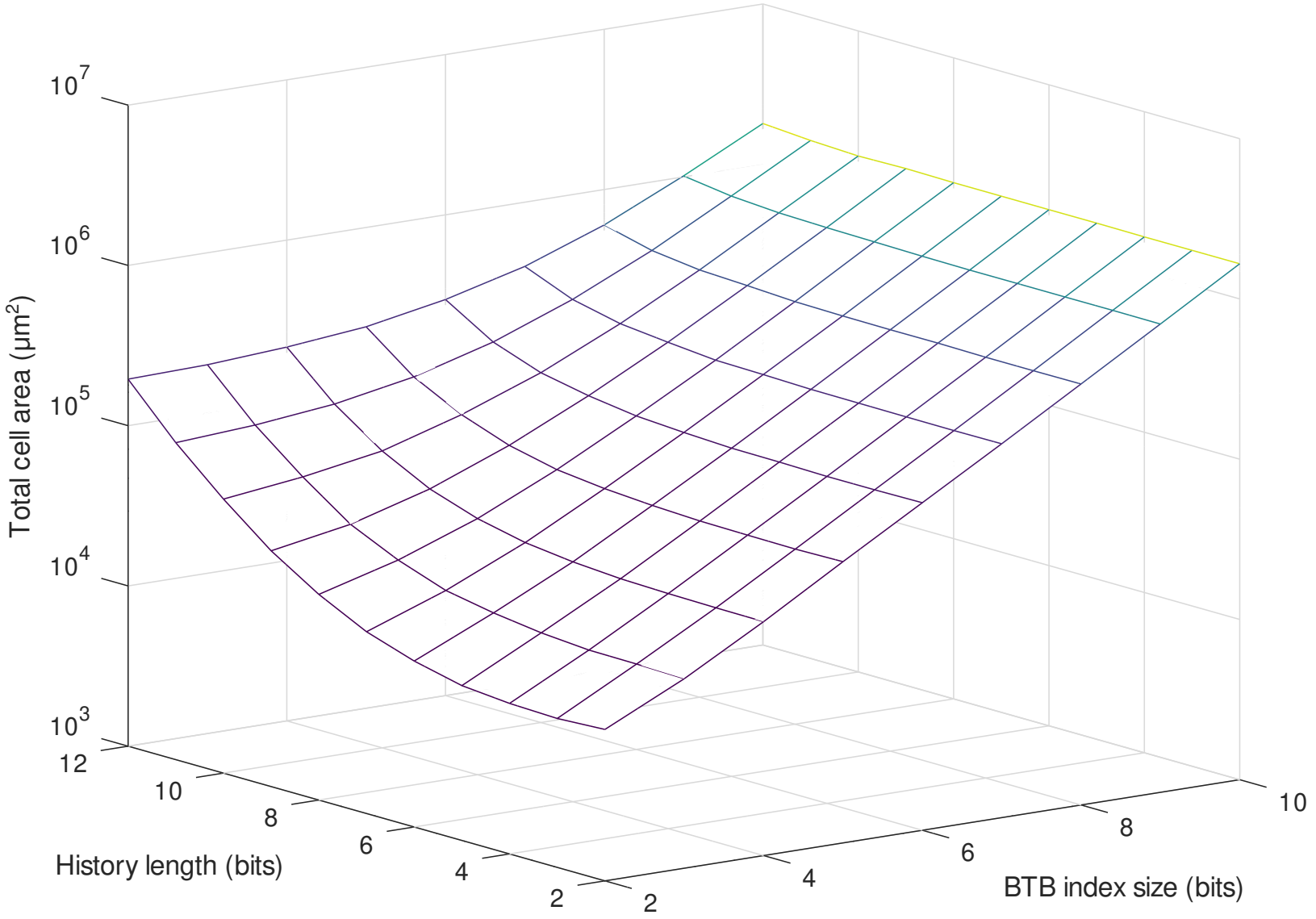
Gshare accuracy vs. History bits



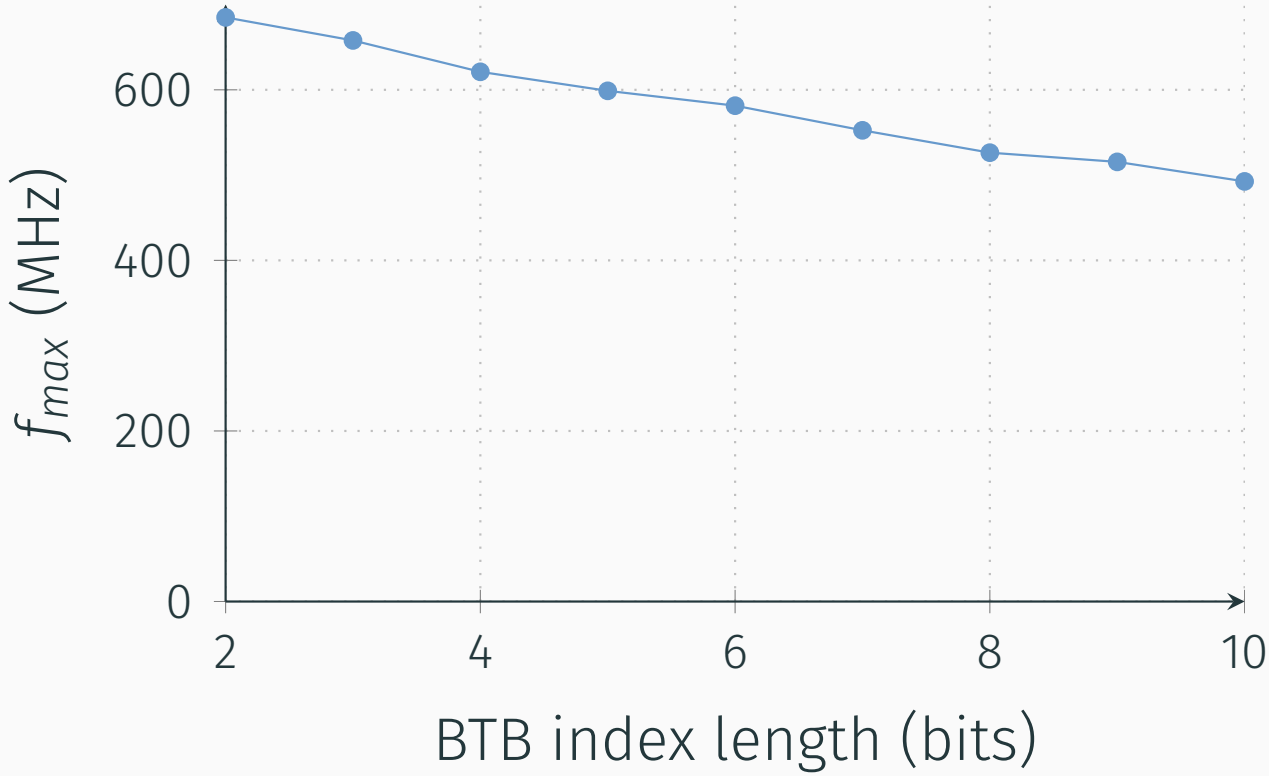
Misprediction penalty vs. BTB size



Synthesis area results



Synthesis frequency results



- Frequency depends only on the size of the BTB
- The BTB address decoding network is the critical path of the whole design

Concluding remarks

- Even if results are not top-class, this exploratory work proved insightful.

Concluding remarks

- Even if results are not top-class, this exploratory work proved insightful.
- Useful in teaching, to explore firsthand a real implementation of an out-of-order processor.

Concluding remarks

- Even if results are not top-class, this exploratory work proved insightful.
- Useful in teaching, to explore firsthand a real implementation of an out-of-order processor.
- Academic potential, thanks to ISA extensions for parallel workloads (e.g. machine learning algorithms).

Thank you for your attention!