Report– Project 1

Course : 431 Intro to Compute Architecture

Name: Hong Zhang, Haoxuan Zhang

PSUID: 965232865, 901437863

1. Method

Given an initial 18–dimensional parameters, we ‘d like to explore the best parameters for some specific object function in the process of designing computer architecture. The 18–dimensional parameter vector is defined in the ‘runprojectsuite.sh’ file like the size of L1 instruction set, choice of branch predictor and so on. An exhaustive DSE simply tries out all possible combinations of parameter values to
find the absolute best design. However it will be very time–consuming. Thus we explore the parameters in a heuristic way. Actually it is a greedy method to find the best value for each dimension one by one. More specifically, we can test all the possible values in one dimension by priority and fix the values in other dimensions. Then select the best local value for the tested dimension. Doing this search method recursively until the performance won’t be improved anymore or it meets one of terminal conditions.

In our case, we set the priority of parameters as [BP, Cache, Core, FPU]. That is, we test all the possible values in the dimensions related to BP first. Then find each local step for each dimension by the order of Cache, Core, FPU. After exploring all the 18 dimensions, go back to the initial state and restart exploring by the latest parameters we get from last iteration.

Two kinds of parameters are explored in order to optimize the two object functions: The “best” performing overall design (in term of the geometric mean of normalized execution time normalized across all benchmarks) and The most energy–efficient design

(as measured by the lowest geometric mean of normalized energy–delay product [units of energy delay product are joule–seconds]
across all benchmarks).

2. List of design point chosed by DSE

To optimize the best execution time, we have 70 design points.

iter	0	1	2	3	4	5	6
width	1	1	1	1	1	1	1
scheduling	issue:inorder true issue:wrong path false	issue:inorder true issue:wrong path false	issue:inorder true issue:wrong path false	issue:inorder true issue:wrong path false	issue:inorder true issue:wrong path false	issue:inorder true issue:wrong path false	issue:inorder true issue:wrong path false
l1block	8	8	8	8	8	8	8
dl1sets	1024	1024	1024	1024	1024	1024	1024
dl1assoc	1	1	1	1	1	1	1
il1sets	1024	1024	1024	1024	1024	1024	1024
il1assoc	1	1	1	1	1	1	1
ul2sets	1024	1024	1024	1024	1024	1024	1024
ul2block	64	64	64	64	64	64	64
ul2assoc	4	4	4	4	4	4	4
replacepolicy	l	l	l	l	l	l	l
fpwidth	2	2	2	2	2	2	2
branchsettings	bpred bimod bpred:bimod 2048	bpred 2lev bpred:2lev 1 1024 8 0	bpred 2lev bpred:2lev 4 256 8 0	bpred comb bpred:comb 1024	bpred comb bpred:comb 1024	bpred comb bpred:comb 1024	bpred comb bpred:comb 1024
ras	2	2	2	2	1	4	8
btb	512 4	512 4	512 4	512 4	512 4	512 4	512 4
dl1lat	3	3	3	3	3	3	3
il1lat	3	3	3	3	3	3	3
ul2lat	10	10	10	10	10	10	10

iter	7	8	9	10	11	12
width	1	1	1	1	1	1
scheduling	issue:inorder true	issue:inorder true	issue:inorder true	issue:inorder true	issue:inorder true	issue:inorder true

	issue:wrongpath false	issue:wrongpath false	issue:wrongpath false	issue:wrongpath false	issue:wrongpath false	issue:wrongpath false
l1block	8	8	8	8	16	32
dl1sets	1024	1024	1024	1024	1024	1024
dl1assoc	1	1	1	1	1	1
il1sets	1024	1024	1024	1024	1024	1024
il1assoc	1	1	1	1	1	1
ul2sets	1024	1024	1024	1024	1024	1024
ul2block	64	64	64	64	64	64
ul2assoc	4	4	4	4	4	4
replacepolicy	l	l	l	l	l	l
fpwidth	2	2	2	2	2	2
branchsettings	bpred comb bpred:comb 1024	bpred comb bpred:comb 1024	bpred comb bpred:comb 1024	bpred comb bpred:comb 1024	bpred comb bpred:comb 1024	bpred comb bpred:comb 1024
ras	8	8	8	8	8	8
btb	128 16	256 8	1024 2	2048 1	2048 1	2048 1
dl1lat	3	3	3	3	4	5
il1lat	3	3	3	3	4	5
ul2lat	10	10	10	10	10	10
iter	13	14	15	16	17	18
width	1	1	1	1	1	1
scheduling	issue:inorder true issue:wrongpath false	issue:inorder true issue:wrongpath false	issue:inorder true issue:wrongpath false	issue:inorder true issue:wrongpath false	issue:inorder true issue:wrongpath false	issue:inorder true issue:wrongpath false
l1block	32	32	32	32	32	32
dl1sets	64	128	256	512	2048	128
dl1assoc	1	1	1	1	1	2
il1sets	1024	1024	1024	1024	1024	1024
il1assoc	1	1	1	1	1	1
ul2sets	1024	1024	1024	1024	1024	1024
ul2block	64	64	64	64	64	64
ul2assoc	4	4	4	4	4	4
replacepolicy	l	l	l	l	l	l
fpwidth	2	2	2	2	2	2
branchsettings	bpred comb bpred:comb 1024	bpred comb bpred:comb 1024	bpred comb bpred:comb 1024	bpred comb bpred:comb 1024	bpred comb bpred:comb 1024	bpred comb bpred:comb 1024
ras	8	8	8	8	8	8
btb	2048 1	2048 1	2048 1	2048 1	2048 1	2048 1

dl1lat	1	2	3	4	6	4
il1lat	5	5	5	5	5	5
ul2lat	10	10	10	10	10	10

iter	19	20	21	22	23	24
width	1	1	1	1	1	1
scheduling	issue:inorder true issue:wrongpath false	issue:inorder true issue:wrongpath false	issue:inorder true issue:wrongpath false	issue:inorder true issue:wrongpath false	issue:inorder true issue:wrongpath false	issue:inorder true issue:wrongpath false
l1block	32	32	32	32	32	32
dl1sets	128	128	128	128	128	128
dl1assoc	4	1	1	1	1	1
il1sets	1024	64	128	256	512	2048
il1assoc	1	1	1	1	1	1
ul2sets	1024	1024	1024	1024	1024	1024
ul2block	64	64	64	64	64	64
ul2assoc	4	4	4	4	4	4
replacepolicy	l	l	l	l	l	l
fpwidth	2	2	2	2	2	2
branchsettings	bpred comb bpred:comb 1024	bpred comb bpred:comb 1024	bpred comb bpred:comb 1024	bpred comb bpred:comb 1024	bpred comb bpred:comb 1024	bpred comb bpred:comb 1024
ras	8	8	8	8	8	8
btb	2048 1	2048 1	2048 1	2048 1	2048 1	2048 1
dl1lat	6	2	2	2	2	2
il1lat	5	1	2	3	4	6
ul2lat	10	10	10	10	10	10

iter	25	26	27	28	29	30
width	1	1	1	1	1	1
scheduling	issue:inorder true issue:wrongpath false	issue:inorder true issue:wrongpath false	issue:inorder true issue:wrongpath false	issue:inorder true issue:wrongpath false	issue:inorder true issue:wrongpath false	issue:inorder true issue:wrongpath false
l1block	32	32	32	32	32	32
dl1sets	128	128	128	128	128	128
dl1assoc	1	1	1	1	1	1
il1sets	2048	2048	2048	2048	2048	2048
il1assoc	1	1	1	1	1	1
ul2sets	2048	4096	1024	1024	1024	1024
ul2block	64	64	128	128	128	128

ul2assoc	4	4	4	2	8	2
replacepolicy	l	l	l	l	l	f
fpwidth	2	2	2	2	2	2
branchsettings	bpred comb bpred:comb 1024	bpred comb bpred:comb 1024	bpred comb bpred:comb 1024	bpred comb bpred:comb 1024	bpred comb bpred:comb 1024	bpred comb bpred:comb 1024
ras	8	8	8	8	8	8
btb	2048 1	2048 1	2048 1	2048 1	2048 1	2048 1
dl1lat	2	2	2	2	2	2
il1lat	6	6	6	6	6	6
ul2lat	11	12	11	9	13	9

iter	31	32	33	34	35	36
width	1	2	4	1	1	1
scheduling	issue:inorder true issue:wrongpath false	issue:inorder true issue:wrongpath false	issue:inorder true issue:wrongpath false	issue:inorder false issue:wrongpath true	issue:inorder true issue:wrongpath false	issue:inorder true issue:wrongpath false
l1block	32	32	32	32	32	32
dl1sets	128	128	128	128	128	128
dl1assoc	1	1	1	1	1	1
il1sets	2048	2048	2048	2048	2048	2048
il1assoc	1	1	1	1	1	1
ul2sets	1024	1024	1024	1024	1024	1024
ul2block	128	128	128	128	128	128
ul2assoc	2	2	2	2	2	2
replacepolicy	r	r	r	r	l	l
fpwidth	2	2	2	2	1	4
branchsettings	bpred comb bpred:comb 1024	bpred comb bpred:comb 1024	bpred comb bpred:comb 1024	bpred comb bpred:comb 1024	bpred comb bpred:comb 1024	bpred comb bpred:comb 1024
ras	8	8	8	8	8	8
btb	2048 1	2048 1	2048 1	2048 1	2048 1	2048 1
dl1lat	2	2	2	2	2	2
il1lat	6	6	6	6	6	6
ul2lat	9	9	9	9	9	9

iter	37	38	39	40	41	42
width	1	1	1	1	1	1
scheduling	issue:inorder true	issue:inorder true	issue:inorder true	issue:inorder true	issue:inorder true	issue:inorder true

	issue:wrongpath false	issue:wrongpath false	issue:wrongpath false	issue:wrongpath false	issue:wrongpath false	issue:wrongpath false
l1block	32	32	32	32	32	32
dl1sets	128	128	128	128	128	128
dl1assoc	1	1	1	1	1	1
il1sets	2048	2048	2048	2048	2048	2048
il1assoc	1	1	1	1	1	1
ul2sets	1024	1024	1024	1024	1024	1024
ul2block	128	128	128	128	128	128
ul2assoc	2	2	2	2	2	2
replacepolicy	l	l	l	l	l	l
fpwidth	8	1	1	1	1	1
branchsettings	bpred comb bpred:comb 1024	bpred nottaken	bpred bimod bpred:bimod 2048	bpred 2lev bpred:2lev 1 1024 8 0	bpred 2lev bpred:2lev 4 256 8 0	bpred comb bpred:comb 1024
ras	8	8	8	8	8	1
btb	2048 1	2048 1	2048 1	2048 1	2048 1	2048 1
dl1lat	2	2	2	2	2	2
il1lat	6	6	6	6	6	6
ul2lat	9	9	9	9	9	9
iter	43	44	45	46	47	48
width	1	1	1	1	1	1
scheduling	issue:inorder true issue:wrongpath false	issue:inorder true issue:wrongpath false	issue:inorder true issue:wrongpath false	issue:inorder true issue:wrongpath false	issue:inorder true issue:wrongpath false	issue:inorder true issue:wrongpath false
l1block	32	32	32	32	32	32
dl1sets	128	128	128	128	128	128
dl1assoc	1	1	1	1	1	1
il1sets	2048	2048	2048	2048	2048	2048
il1assoc	1	1	1	1	1	1
ul2sets	1024	1024	1024	1024	1024	1024
ul2block	128	128	128	128	128	128
ul2assoc	2	2	2	2	2	2
replacepolicy	l	l	l	l	l	l
fpwidth	1	1	1	1	1	1
branchsettings	bpred comb bpred:comb 1024	bpred comb bpred:comb 1024	bpred comb bpred:comb 1024	bpred comb bpred:comb 1024	bpred comb bpred:comb 1024	bpred comb bpred:comb 1024
ras	2	4	8	8	8	8
btb	2048 1	2048 1	128 16	256 8	512 4	1024 2

dl1lat	2	2	2	2	2	2
il1lat	6	6	6	6	6	6
ul2lat	9	9	9	9	9	9

iter	49	50	51	52	53	54
width	1	1	1	1	1	1
scheduling	issue:inorder true issue:wrongpath false	issue:inorder true issue:wrongpath false	issue:inorder true issue:wrongpath false	issue:inorder true issue:wrongpath false	issue:inorder true issue:wrongpath false	issue:inorder true issue:wrongpath false
l1block	16	32	32	32	32	32
dl1sets	128	64	256	512	1024	2048
dl1assoc	1	1	1	1	1	1
il1sets	2048	2048	2048	2048	2048	2048
il1assoc	1	1	1	1	1	1
ul2sets	1024	1024	1024	1024	1024	1024
ul2block	128	128	128	128	128	128
ul2assoc	2	2	2	2	2	2
replacepolicy	l	l	l	l	l	l
fpwidth	1	1	1	1	1	1
branchsettings	bpred comb bpred:comb 1024	bpred comb bpred:comb 1024	bpred comb bpred:comb 1024	bpred comb bpred:comb 1024	bpred comb bpred:comb 1024	bpred comb bpred:comb 1024
ras	8	8	8	8	8	8
btb	2048 1	2048 1	2048 1	2048 1	2048 1	2048 1
dl1lat	1	1	3	4	5	6
il1lat	5	6	6	6	6	6
ul2lat	9	9	9	9	9	9

iter	55	56	57	58	59	60
width	1	1	1	1	1	1
scheduling	issue:inorder true issue:wrongpath false	issue:inorder true issue:wrongpath false	issue:inorder true issue:wrongpath false	issue:inorder true issue:wrongpath false	issue:inorder true issue:wrongpath false	issue:inorder true issue:wrongpath false
l1block	32	32	32	32	32	32
dl1sets	128	128	128	128	128	128
dl1assoc	2	4	1	1	1	1
il1sets	2048	2048	64	128	256	512
il1assoc	1	1	1	1	1	1
ul2sets	1024	1024	1024	1024	1024	1024
ul2block	128	128	128	128	128	128

ul2assoc	2	2	2	2	2	2
replacepolicy	l	l	l	l	l	l
fpwidth	1	1	1	1	1	1
branchsettings	bpred comb bpred:comb 1024	bpred comb bpred:comb 1024	bpred comb bpred:comb 1024	bpred comb bpred:comb 1024	bpred comb bpred:comb 1024	bpred comb bpred:comb 1024
ras	8	8	8	8	8	8
btb	2048 1	2048 1	2048 1	2048 1	2048 1	2048 1
dl1lat	4	6	2	2	2	2
il1lat	6	6	1	2	3	4
ul2lat	9	9	9	9	9	9

iter	61	62	63	64	65	66
width	1	1	1	1	1	1
scheduling	issue:inorder true issue:wrongpath false	issue:inorder true issue:wrongpath false	issue:inorder true issue:wrongpath false	issue:inorder true issue:wrongpath false	issue:inorder true issue:wrongpath false	issue:inorder true issue:wrongpath false
l1block	32	32	32	32	32	32
dl1sets	128	128	128	128	128	128
dl1assoc	1	1	1	1	1	1
il1sets	1024	2048	2048	2048	2048	2048
il1assoc	1	1	1	1	1	1
ul2sets	1024	2048	4096	1024	1024	1024
ul2block	128	128	128	128	128	128
ul2assoc	2	2	2	4	8	2
replacepolicy	l	l	l	l	l	f
fpwidth	1	1	1	1	1	1
branchsettings	bpred comb bpred:comb 1024	bpred comb bpred:comb 1024	bpred comb bpred:comb 1024	bpred comb bpred:comb 1024	bpred comb bpred:comb 1024	bpred comb bpred:comb 1024
ras	8	8	8	8	8	8
btb	2048 1	2048 1	2048 1	2048 1	2048 1	2048 1
dl1lat	2	2	2	2	2	2
il1lat	5	6	6	6	6	6
ul2lat	9	10	11	11	13	9

iter	67	68	69	70
width	1	2	4	1
scheduling	issue:inorder true	issue:inorder true	issue:inorder true	issue:inorder false

	issue:wrongpath false	issue:wrongpath false	issue:wrongpath false	issue:wrongpath true
l1block	32	32	32	32
dl1sets	128	128	128	128
dl1assoc	1	1	1	1
il1sets	2048	2048	2048	2048
il1assoc	1	1	1	1
ul2sets	1024	1024	1024	1024
ul2block	128	128	128	128
ul2assoc	2	2	2	2
replacepolicy	r	r	r	r
fpwidth	1	1	1	1
branchsettings	bpred comb bpred:comb 1024	bpred comb bpred:comb 1024	bpred comb bpred:comb 1024	bpred comb bpred:comb 1024
ras	8	8	8	8
btb	2048 1	2048 1	2048 1	2048 1
dl1lat	2	2	2	2
il1lat	6	6	6	6
ul2lat	9	9	9	9

To explore energy efficient, we have 73 design points.

iter	0	1	2	3	4	5
width	1	1	1	1	1	1
scheduling	issue:inorder true issue:wrongpath false	issue:inorder true issue:wrongpath false	issue:inorder true issue:wrongpath false	issue:inorder true issue:wrongpath false	issue:inorder true issue:wrongpath false	issue:inorder true issue:wrongpath false
l1block	8	8	8	8	8	8
dl1sets	1024	1024	1024	1024	1024	1024
dl1assoc	1	1	1	1	1	1
il1sets	1024	1024	1024	1024	1024	1024
il1assoc	1	1	1	1	1	1
ul2sets	1024	1024	1024	1024	1024	1024
ul2block	64	64	64	64	64	64
ul2assoc	4	4	4	4	4	4
replacepolicy	l	l	l	l	l	l
fpwidth	2	2	2	2	2	2

branchsettings	bpred bimod bpred:bimod 2048	bpred 2lev bpred:2lev 1 1024 8 0	bpred 2lev bpred:2lev 4 256 8 0	bpred comb bpred:comb 1024	bpred comb bpred:comb 1024	bpred comb bpred:comb 1024
ras	2	2	2	2	1	4
btb	512 4	512 4	512 4	512 4	512 4	512 4
dl1lat	3	3	3	3	3	3
il1lat	3	3	3	3	3	3
ul2lat	10	10	10	10	10	10
iter	6	7	8	9	10	11
width	1	1	1	1	1	1
scheduling	issue:inorder true issue:wrongpath false	issue:inorder true issue:wrongpath false	issue:inorder true issue:wrongpath false	issue:inorder true issue:wrongpath false	issue:inorder true issue:wrongpath false	issue:inorder true issue:wrongpath false
l1block	8	8	8	8	8	16
dl1sets	1024	1024	1024	1024	1024	1024
dl1assoc	1	1	1	1	1	1
il1sets	1024	1024	1024	1024	1024	1024
il1assoc	1	1	1	1	1	1
ul2sets	1024	1024	1024	1024	1024	1024
ul2block	64	64	64	64	64	64
ul2assoc	4	4	4	4	4	4
replacepolicy	l	l	l	l	l	l
fpwidth	2	2	2	2	2	2
branchsettings	bpred comb bpred:comb 1024	bpred comb bpred:comb 1024	bpred comb bpred:comb 1024	bpred comb bpred:comb 1024	bpred comb bpred:comb 1024	bpred comb bpred:comb 1024
ras	8	8	8	8	8	8
btb	512 4	128 16	256 8	1024 2	2048 1	2048 1
dl1lat	3	3	3	3	3	4
il1lat	3	3	3	3	3	4
ul2lat	10	10	10	10	10	10
iter	12	13	14	15	16	17
width	1	1	1	1	1	1
scheduling	issue:inorder true	issue:inorder true	issue:inorder true	issue:inorder true	issue:inorder true	issue:inorder true

	issue:wrongpath h false	issue:wrongpath h false	issue:wrongpath h false	issue:wrongpath h false	issue:wrongpath h false	issue:wrongpath h false
l1block	32	32	32	32	32	32
dl1sets	1024	64	128	256	512	2048
dl1assoc	1	1	1	1	1	1
il1sets	1024	1024	1024	1024	1024	1024
il1assoc	1	1	1	1	1	1
ul2sets	1024	1024	1024	1024	1024	1024
ul2block	64	64	64	64	64	64
ul2assoc	4	4	4	4	4	4
replacepolicy	l	l	l	l	l	l
fpwidth	2	2	2	2	2	2
branchsettings	bpred comb bpred:comb 1024	bpred comb bpred:comb 1024	bpred comb bpred:comb 1024	bpred comb bpred:comb 1024	bpred comb bpred:comb 1024	bpred comb bpred:comb 1024
ras	8	8	8	8	8	8
btb	2048 1	2048 1	2048 1	2048 1	2048 1	2048 1
dl1lat	5	1	2	3	4	6
il1lat	5	5	5	5	5	5
ul2lat	10	10	10	10	10	10

iter	18	19	20	21	22	23	24	25
width	1	1	1	1	1	1	1	1

scheduling	issue:inorder true issue:wrongpath false	issue:inorder true issue:wrongpath false	issue:inorder true issue:wrongpath false	issue:inorder true issue:wrongpath false	issue:inorder true issue:wrongpath false	issue:inorder true issue:wrongpath false	issue:inorder true issue:wrongpath false	issue:inorder true issue:wrongpath false
l1block	32	32	32	32	32	32	32	32
dl1sets	128	128	128	128	128	128	128	128
dl1assoc	2	4	1	1	1	1	1	1
il1sets	1024	1024	64	128	256	512	2048	1024
il1assoc	1	1	1	1	1	1	1	2
ul2sets	1024	1024	1024	1024	1024	1024	1024	1024
ul2block	64	64	64	64	64	64	64	64
ul2assoc	4	4	4	4	4	4	4	4
replacepolicy	l	l	l	l	l	l	l	l
fpwidth	2	2	2	2	2	2	2	2

branchsettings	bpred comb bpred:comb 1024	bpred comb bpred:comb 1024	bpred comb bpred:comb 1024	bpred comb bpred:comb 1024	bpred comb bpred:comb 1024	bpred comb bpred:comb 1024	bpred comb bpred:comb 1024	bpred comb bpred:comb 1024
ras	8	8	8	8	8	8	8	8
btb	2048 1	2048 1	2048 1	2048 1	2048 1	2048 1	2048 1	2048 1
dl1lat	4	6	2	2	2	2	2	2
il1lat	5	5	1	2	3	4	6	7
ul2lat	10	10	10	10	10	10	10	10

iter	26	27	28	29	30	31
width	1	1	1	1	1	1
scheduling	issue:inorder true issue:wrongpath false	issue:inorder true issue:wrongpath false	issue:inorder true issue:wrongpath false	issue:inorder true issue:wrongpath false	issue:inorder true issue:wrongpath false	issue:inorder true issue:wrongpath false
l1block	32	32	32	32	32	32
dl1sets	128	128	128	128	128	128
dl1assoc	1	1	1	1	1	1
il1sets	1024	1024	1024	1024	1024	1024
il1assoc	1	1	1	1	1	1
ul2sets	512	2048	4096	512	512	512
ul2block	64	64	64	128	128	128
ul2assoc	4	4	4	4	2	8
replacepolicy	l	l	l	l	l	l
fpwidth	2	2	2	2	2	2
branchsettings	bpred comb bpred:comb 1024	bpred comb bpred:comb 1024	bpred comb bpred:comb 1024	bpred comb bpred:comb 1024	bpred comb bpred:comb 1024	bpred comb bpred:comb 1024
ras	8	8	8	8	8	8
btb	2048 1	2048 1	2048 1	2048 1	2048 1	2048 1
dl1lat	2	2	2	2	2	2
il1lat	5	5	5	5	5	5
ul2lat	9	11	12	10	8	12

iter	32	33	34	35	36	37
width	1	1	1	2	4	1
scheduling	issue:inorder true issue:wrongpath false	issue:inorder true issue:wrongpath false	issue:inorder true issue:wrongpath false	issue:inorder true issue:wrongpath false	issue:inorder true issue:wrongpath false	issue:inorder false issue:wrongpath true
l1block	32	32	32	32	32	32
dl1sets	128	128	128	128	128	128
dl1assoc	1	1	1	1	1	1
il1sets	1024	1024	1024	1024	1024	1024
il1assoc	1	1	1	1	1	1
ul2sets	512	512	512	512	512	512
ul2block	128	128	128	128	128	128
ul2assoc	16	2	2	2	2	2
replacepolicy	l	f	r	r	r	r
fpwidth	2	2	2	2	2	2
branchsettings	bpred comb bpred:comb 1024	bpred comb bpred:comb 1024	bpred comb bpred:comb 1024	bpred comb bpred:comb 1024	bpred comb bpred:comb 1024	bpred comb bpred:comb 1024
ras	8	8	8	8	8	8
btb	2048 1	2048 1	2048 1	2048 1	2048 1	2048 1
dl1lat	2	2	2	2	2	2
il1lat	5	5	5	5	5	5
ul2lat	14	8	8	8	8	8

iter	38	39	40	41	42	43
width	1	1	1	1	1	1
scheduling	issue:inorder true issue:wrongpath false	issue:inorder true issue:wrongpath false	issue:inorder true issue:wrongpath false	issue:inorder true issue:wrongpath false	issue:inorder true issue:wrongpath false	issue:inorder true issue:wrongpath false
l1block	32	32	32	32	32	32
dl1sets	128	128	128	128	128	128
dl1assoc	1	1	1	1	1	1
il1sets	1024	1024	1024	1024	1024	1024
il1assoc	1	1	1	1	1	1
ul2sets	512	512	512	512	512	512

ul2block	128	128	128	128	128	128
ul2assoc	2	2	2	2	2	2
replacepolicy	l	l	l	l	l	l
fpwidth	1	4	8	1	1	1
branchsettings	bpred comb bpred:comb 1024	bpred comb bpred:comb 1024	bpred comb bpred:comb 1024	bpred nottaken	bpred bimod bpred:bimod 2048	bpred 2lev bpred:2lev 1 1024 8 0
ras	8	8	8	8	8	8
btb	2048 1	2048 1	2048 1	2048 1	2048 1	2048 1
dl1lat	2	2	2	2	2	2
il1lat	5	5	5	5	5	5
ul2lat	8	8	8	8	8	8

iter	44	45	46	47	48	49
width	1	1	1	1	1	1
scheduling	issue:inorder true issue:wrongpath false	issue:inorder true issue:wrongpath false	issue:inorder true issue:wrongpath false	issue:inorder true issue:wrongpath false	issue:inorder true issue:wrongpath false	issue:inorder true issue:wrongpath false
l1block	32	32	32	32	32	32
dl1sets	128	128	128	128	128	128
dl1assoc	1	1	1	1	1	1
il1sets	1024	1024	1024	1024	1024	1024
il1assoc	1	1	1	1	1	1
ul2sets	512	512	512	512	512	512
ul2block	128	128	128	128	128	128
ul2assoc	2	2	2	2	2	2
replacepolicy	l	l	l	l	l	l
fpwidth	1	1	1	1	1	1
branchsettings	bpred 2lev bpred:2lev 4 256 8 0	bpred comb bpred:comb 1024	bpred comb bpred:comb 1024	bpred comb bpred:comb 1024	bpred comb bpred:comb 1024	bpred comb bpred:comb 1024
ras	8	1	2	4	8	8
btb	2048 1	2048 1	2048 1	2048 1	128 16	256 8
dl1lat	2	2	2	2	2	2
il1lat	5	5	5	5	5	5
ul2lat	8	8	8	8	8	8

iter	50	51	52	53	54	55

width	1	1	1	1	1	1
scheduling	issue:inorder true issue:wrongpath false	issue:inorder true issue:wrongpath false	issue:inorder true issue:wrongpath false	issue:inorder true issue:wrongpath false	issue:inorder true issue:wrongpath false	issue:inorder true issue:wrongpath false
l1block	32	32	16	32	32	32
dl1sets	128	128	128	64	256	512
dl1assoc	1	1	1	1	1	1
il1sets	1024	1024	1024	1024	1024	1024
il1assoc	1	1	1	1	1	1
ul2sets	512	512	512	512	512	512
ul2block	128	128	128	128	128	128
ul2assoc	2	2	2	2	2	2
replacepolicy	l	l	l	l	l	l
fpwidth	1	1	1	1	1	1
branchsettings	bpred comb bpred:comb 1024	bpred comb bpred:comb 1024	bpred comb bpred:comb 1024	bpred comb bpred:comb 1024	bpred comb bpred:comb 1024	bpred comb bpred:comb 1024
ras	8	8	8	8	8	8
btb	512 4	1024 2	2048 1	2048 1	2048 1	2048 1
dl1lat	2	2	1	1	3	4
il1lat	5	5	4	5	5	5
ul2lat	8	8	8	8	8	8

iter	56	57	58	59	60	61
width	1	1	1	1	1	1
scheduling	issue:inorder true issue:wrongpath false	issue:inorder true issue:wrongpath false	issue:inorder true issue:wrongpath false	issue:inorder true issue:wrongpath false	issue:inorder true issue:wrongpath false	issue:inorder true issue:wrongpath false
l1block	32	32	32	32	32	32
dl1sets	1024	128	128	128	128	128
dl1assoc	1	2	4	1	1	1
il1sets	1024	1024	1024	64	128	256
il1assoc	1	1	1	1	1	1
ul2sets	512	512	512	512	512	512
ul2block	128	128	128	128	128	128
ul2assoc	2	2	2	2	2	2
replacepolicy	l	l	l	l	l	l
fpwidth	1	1	1	1	1	1

branchsettings	bpred comb bpred:comb 1024	bpred comb bpred:comb 1024	bpred comb bpred:comb 1024	bpred comb bpred:comb 1024	bpred comb bpred:comb 1024	bpred comb bpred:comb 1024
ras	8	8	8	8	8	8
btb	2048 1	2048 1	2048 1	2048 1	2048 1	2048 1
dl1lat	5	4	6	2	2	2
il1lat	5	5	5	1	2	3
ul2lat	8	8	8	8	8	8
iter	62	63	64	65	66	67
width	1	1	1	1	1	1
scheduling	issue:inorder true issue:wrongpath false	issue:inorder true issue:wrongpath false	issue:inorder true issue:wrongpath false	issue:inorder true issue:wrongpath false	issue:inorder true issue:wrongpath false	issue:inorder true issue:wrongpath false
l1block	32	32	32	32	32	32
dl1sets	128	128	128	128	128	128
dl1assoc	1	1	1	1	1	1
il1sets	512	1024	1024	1024	1024	1024
il1assoc	1	1	1	1	1	1
ul2sets	512	1024	2048	4096	512	512
ul2block	128	128	128	128	128	128
ul2assoc	2	2	2	2	4	8
replacepolicy	l	l	l	l	l	l
fpwidth	1	1	1	1	1	1
branchsettings	bpred comb bpred:comb 1024	bpred comb bpred:comb 1024	bpred comb bpred:comb 1024	bpred comb bpred:comb 1024	bpred comb bpred:comb 1024	bpred comb bpred:comb 1024
ras	8	8	8	8	8	8
btb	2048 1	2048 1	2048 1	2048 1	2048 1	2048 1
dl1lat	2	2	2	2	2	2
il1lat	4	5	5	5	5	5
ul2lat	8	9	10	11	10	12
iter	68	69	70	71	72	73
width	1	1	1	2	4	1
scheduling	issue:inorder true issue:wrongpath false	issue:inorder true issue:wrongpath false	issue:inorder true issue:wrongpath false	issue:inorder true issue:wrongpath false	issue:inorder true issue:wrongpath false	issue:inorder false issue:wrongpath true

l1block	32	32	32	32	32	32
dl1sets	128	128	128	128	128	128
dl1assoc	1	1	1	1	1	1
il1sets	1024	1024	1024	1024	1024	1024
il1assoc	1	1	1	1	1	1
ul2sets	512	512	512	512	512	512
ul2block	128	128	128	128	128	128
ul2assoc	16	2	2	2	2	2
replacepolicy	l	f	r	r	r	r
fpwidth	1	1	1	1	1	1
branchsettings	bpred comb bpred:comb 1024	bpred comb bpred:comb 1024	bpred comb bpred:comb 1024	bpred comb bpred:comb 1024	bpred comb bpred:comb 1024	bpred comb bpred:comb 1024
ras	8	8	8	8	8	8
btb	2048 1	2048 1	2048 1	2048 1	2048 1	2048 1
dl1lat	2	2	2	2	2	2
il1lat	5	5	5	5	5	5
ul2lat	14	8	8	8	8	8

3. Table

Select the best parameters for Performance and EDP as the following:

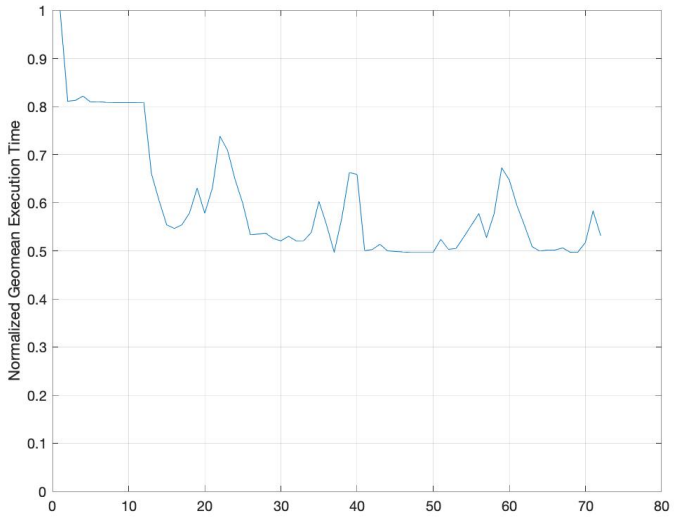
Parameters	Performance	EDP
width	Value: 1 Reason: More instruction pipeline increases instruction latency hence to choose a less instruction pipeline value will improve performance.	Value: 1 Reason: More instruction pipeline is more hard to make a correct prediction, it means higher energy consume. Hence the less instruction pipeline can save energy.
scheduling	Value: issue:inorder true issue:wrongpath false Reason: Under In-order instruction execution, instructions are statically scheduled by the hardware. It means faster clock cycle and improves the performance.	Value: issue:inorder true issue:wrongpath false Reason: Under In-order instruction execution, instructions are statically scheduled by the hardware. It means fewer transistors and saves the energy.
l1block	Value:32 Reason: Larger block size means a greater bandwidth efficiency and improves the accuracy of prediction. It means it can perform faster.	Value:32 Reason: Larger block size means a greater bandwidth efficiency and improves the accuracy of prediction. It means less energy consume.
dl1sets	Value:32 Reason: A block of memory replacement is restricted to a set of	Value:32 Reason: A block of memory replacement is restricted to a set of cache lines by cache

	cache lines by cache replacement policy. To have less sets means higher hit rate when it try to find the target block. In sum, less set improves the performance.	replacement policy. To have less sets means higher hit rate when it try to find the target block. In sum, less set saves energy of finding the correct target block.
dl1assoc	Value:1 Reason: One way set-associative is also called direct mapped. It has less comparators and multiplexer. So, it can perform faster.	Value:1 Reason: One way set-associative is also called direct mapped. It has less comparators and multiplexer. So, the energy consume will be less.
il1sets	Value:2048 Reason: A block of memory replacement is restricted to a set of cache lines by cache replacement policy. To have less sets means higher hit rate when it try to find the target block. In sum, less set improves the performance.	Value:1024 Reason: A block of memory replacement is restricted to a set of cache lines by cache replacement policy. To have less sets means higher hit rate when it try to find the target block. In sum, less set saves energy of finding the correct target block.
il1assoc	Value:1 Reason: One way set-associative is also called direct mapped. It has less comparators and multiplexer. So, it can perform faster.	Value:1 Reason: One way set-associative is also called direct mapped. It has less comparators and multiplexer. So, the energy consume will be less
ul2sets	Value:1024 Reason: A block of memory replacement is restricted to a set of cache lines by cache replacement policy. To have less sets means higher hit rate when it try to find the target block. In sum, less set improves the performance.	Value:512 Reason: A block of memory replacement is restricted to a set of cache lines by cache replacement policy. To have less sets means higher hit rate when it try to find the target block. In sum, less set saves energy of finding the correct target block.
ul2block	Value:128 Reason: Larger block size means a greater bandwidth efficiency and improves the accuracy of prediction. It means it can perform faster.	Value:128 Reason: Larger block size means a greater bandwidth efficiency and improves the accuracy of prediction. It means it saves energy.
ul2assoc	Value:2 Reason: Two way set-associative has relatively less comparators and multiplexer and relatively high hit rate. So, it improves performance.	Value:2 Reason: Two way set-associative has relatively less comparators and multiplexer and relatively high hit rate. So, it saves energy.
replacepolicy	Value:l Reason: LRU replaces the page which is least recently used, it increases the hit rate and improves the performance.	Value:l Reason: LRU replaces the page which is least recently used, it increases the hit rate and saves energy.
fpwidth	Value:1 Reason: Wider floating bit width	Value:1 Reason: Wider floating bit width increases

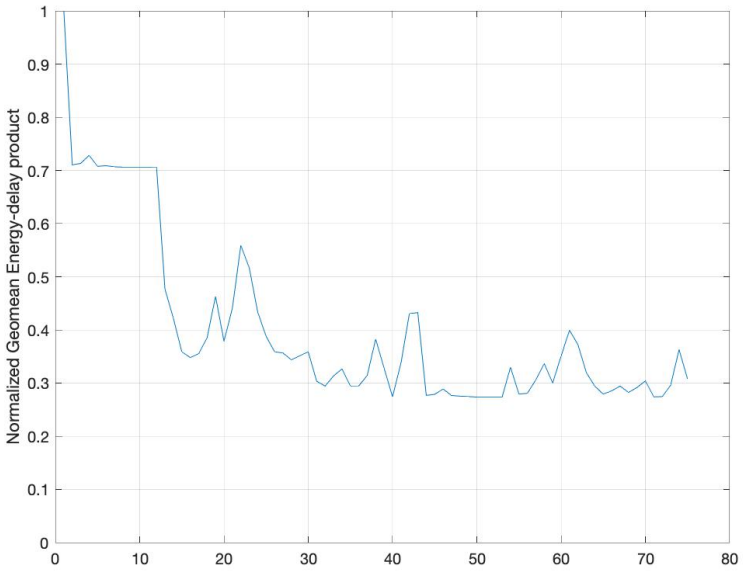
	increases accuracy by requiring more FP multiplication, it means low performance.	accuracy by requiring more FP multiplication, it means high energy consume.
branchsettings	Value: bpred comb bpred:comb 1024 Reason: A combination of Branch prediction can achieve a better accuracy on prediction. It means high performance.	Value: bpred comb bpred:comb 1024 Reason: A combination of Branch prediction can achieve a better accuracy on prediction. It means low energy consume.
ras	Value:8 Reason: Larger stack size means more variable can be store and has a faster response reaction of return the address. So, it increases the performance	Value:8 Reason: Larger stack size means more variable can be store and has a faster response reaction of return the address. So, it saves energy.
btb	Value: 128 16 Reason: It has relatively high hit rate with a relatively low energy consume to maintain.	Value:1024 2 Reason: It has relatively high hit rate with a relatively low energy consume to maintain.

4. Plots

1. plot the value of normalized geomean execution time (NGET) during the design space exploration for the best performant design. In the file '/logs/ExecuteTime.log', we draw all the geomean excution time (NGET) as the y-axis.

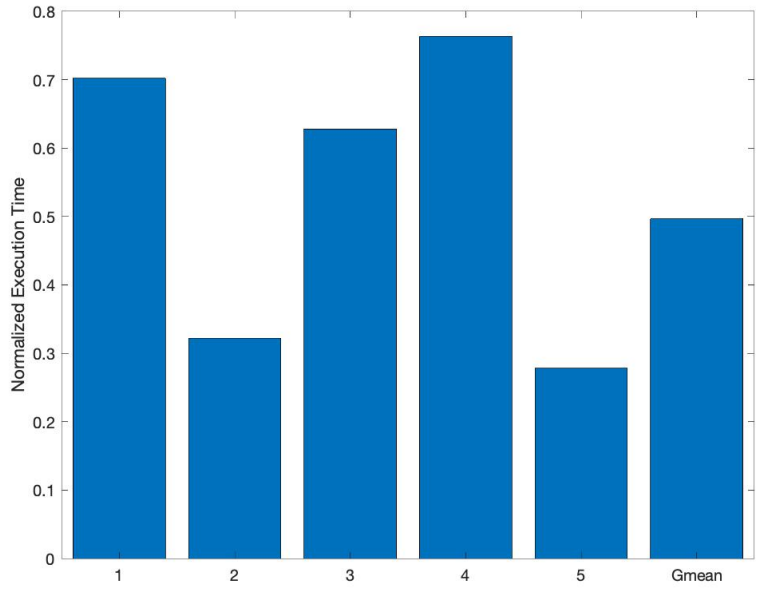


2. normalized geomean of energy-delay product.In the file ‘/logs/EnergyEfficiency.log’, we draw all the normalized geomean of energy-delay product as the y-axis.



3. Bar chart showing normalized per–benchmark execution time and geomean normalized execution time for the best performing design. In the ‘/logs/ExecuteTime.best’, bestPerformantConfig, GeomeanEDPNorm, geomeanExecTimeNorm, GeoEDP, GeoTime, [benchmarkExecTime, benchmarkNormalizedExecTime] (repeated 5 times).

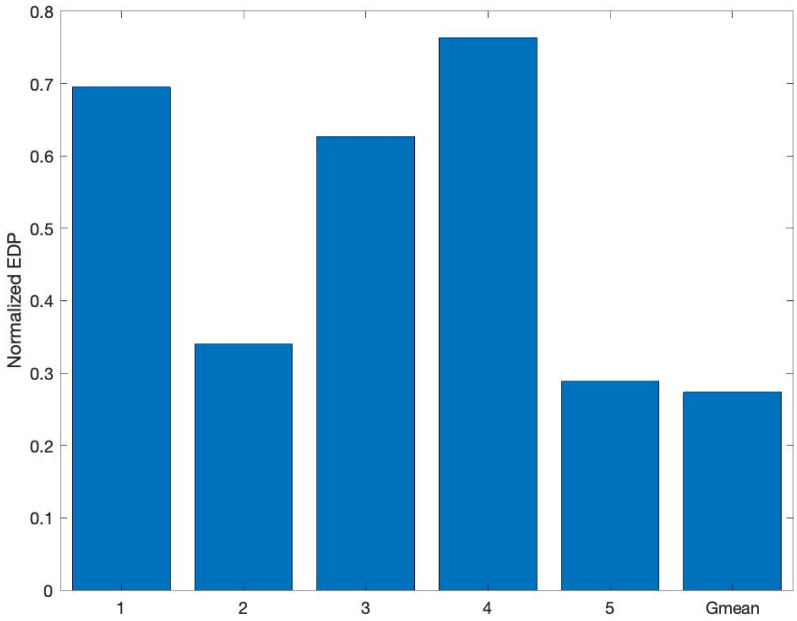
We select 5 benchmarkNormalizedExecTime as the x label (1,2,3,4,5), and compare them to the geomeanExecTimeNorm.



4.Bar chart showing per-benchmark normalized energy-delay product and geomean normalized energy delay product for the most energy-effcient design found. In the ‘/logs/EnergyEfficiency.best’, bestEDPConfig,

GeomeanEDPNorm, geomeanExecTimeNorm, GeoEDP, GeoTime, [benchmarkEDP, benchmarkNormalizedEDP] (repeated 5 times) are presented.

We compare the benchmarkNormalizedEDP to GeomeanEDPNorm as the following bar chart.



5. Instead of testing all the 18 dimensional parameter, maybe we can fix some parameters in advance by the prior knowledge. For example, we always chose a better branch prediction method. Besides, maybe we can chose more initial state to find the best parameters locally, then select the optimal one.

6. Insights gained from this project

1. This method will usually give us a local optimal by testing different value one by one dimension separately. However, there are some prior knowledge gained from computer architecture that will be helpful for us to design the parameters. For example, usually more ALUs may increase performance by extracting more ILP. And a better design Branch Prediction method will be helpful to improve it too.

2. It doesn't means that the large size of L1 cache or block size or more ALUs can promise a better performance. Parameters are affected to each other. We can't hope to get a better performance by simply enhancing the volume of hardware.