信息科学与技术学院
School of Information Science and Technology

CS 110
Computer Architecture
Datapath II

Instructors: Siting Liu & Yuan Xiao
Course website: https://faculty.sist.shanghaitech.edu.cn/liust/courses/CS110.html
School of Information Science and Technology (SIST)
ShanghaiTech University

2026/04/09

Administratives

- Lab 6 released.
- HW 4 released, ddl April 21st! Start early!
- HW 3 DDL TODAY!
- Project 1.2 released, DDL April 27th.
- Discussion (SPST 4-122 18:00 - 19:40) schedule
  - Apr. 10th on digital circuit by Lehan Liu;
  - Apr. 17th on mid-term I review by Chaofan Li;
  - Apr. 24th on datapath by Yuxuan Li.

Mid-term I

- Midterm I
  - April 23rd 8:00 am - 10:00 am
    - We start sharp at 8:00 am!
    - Arrive 7:45 am to check-in (Venue: TBD on your egate system; Seat: TBD on-site)
    - Arrive later then 8:30 am will get 0 mark!
- Contents:
  - Everything till April 21st lecture (included)
- Switch cell phones **off**!!! (not silent mode)
  - **Put them in your bags**.
- **Bags in the front**. On the table: nothing but **pen**, exam paper, 1 drink, 1 snack, **your student ID card** and **your cheat sheet**!

Mid-term I requirements

- You can bring a cheatsheet (**handwritten only**). **1**-page **A4**, **double-sided** (2-page for the mid-term II and 3-page for the final). Put it on your desk at exam. Cheatsheet that does not apply to the rules **would be taken away**.
- Greencard shown on the course website is **provided** with the exam paper.
- No other electronic devices are allowed!
  - No ear plugs, music, smartwatch, calculator, computer…
- Anybody touching any electronic device will **FAIL** the course!
- Anybody found cheating (copy your neighbors answers, additional material, ...) will **FAIL** the course!


CASIO
JF-120TM
12 DIGITS
TWO WAY POWER
F CUT 5/4
4 3 2 1 0 A
+/-
MRC
C
AC
ON
7
4
8
0
00
1
9
6
3
=
X
TAX-
TAX+
TAX RATE
RATE SET
%
GT

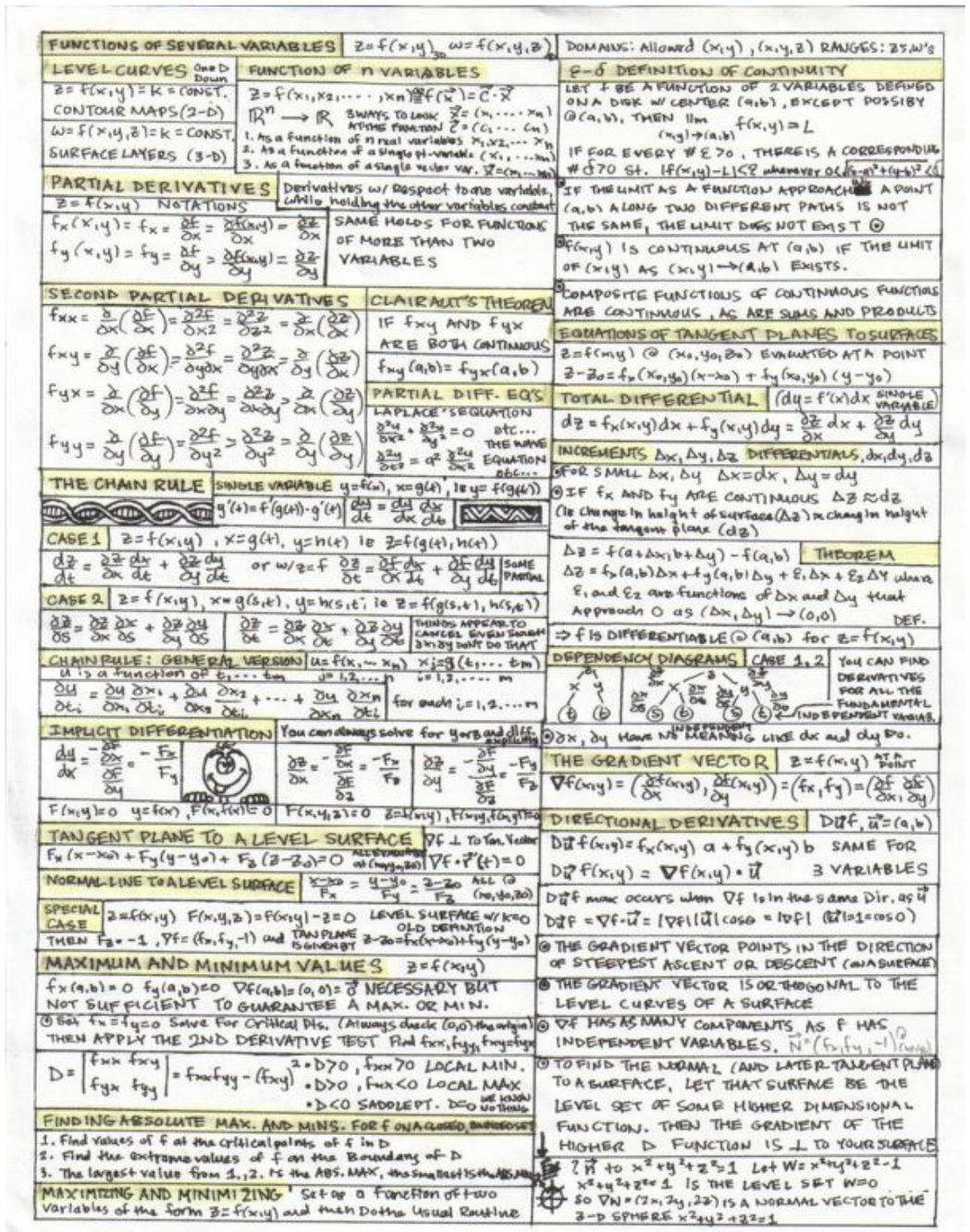

08:08
Thu, Sep 1
HUAWEI

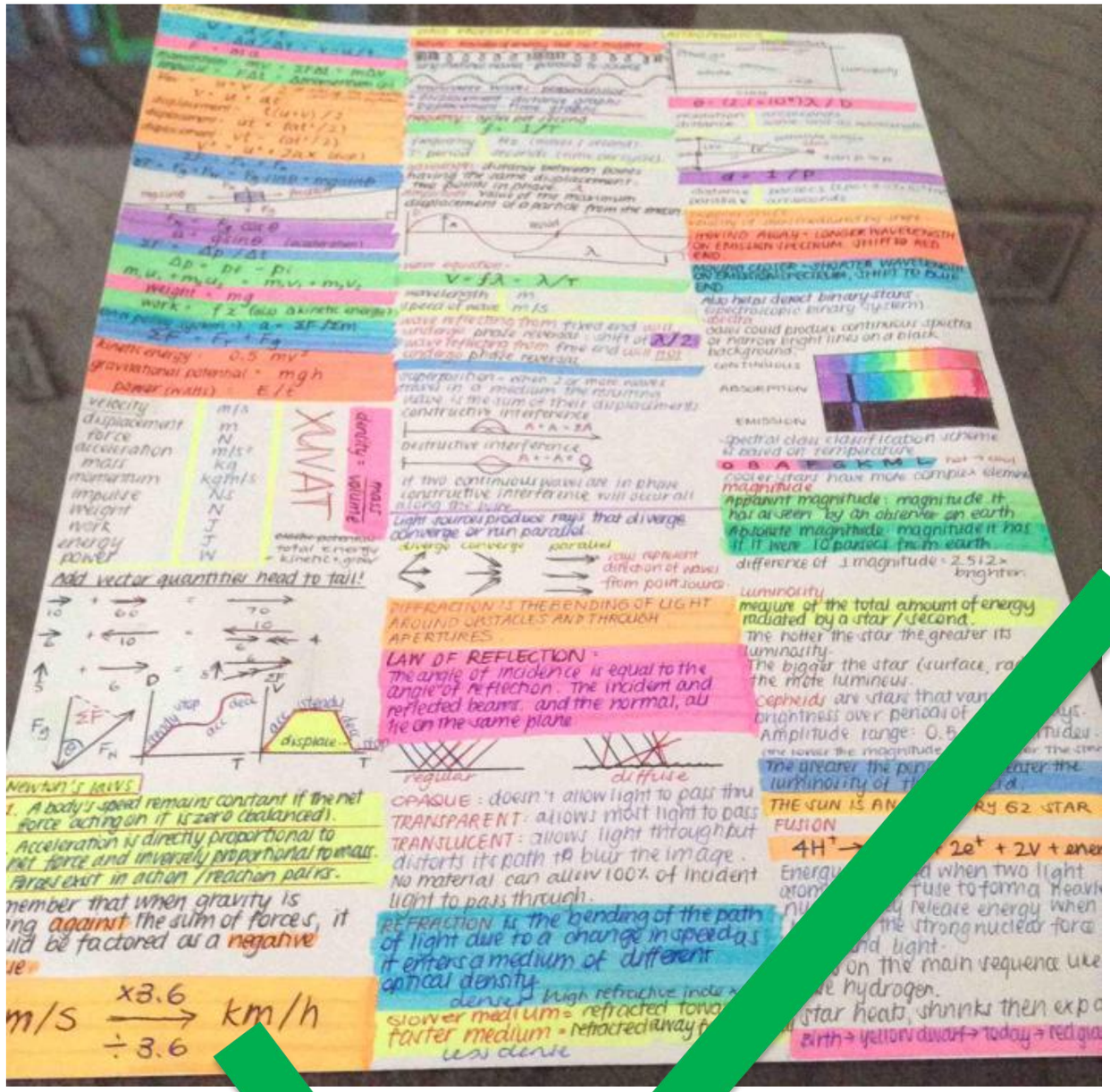

Tue
9

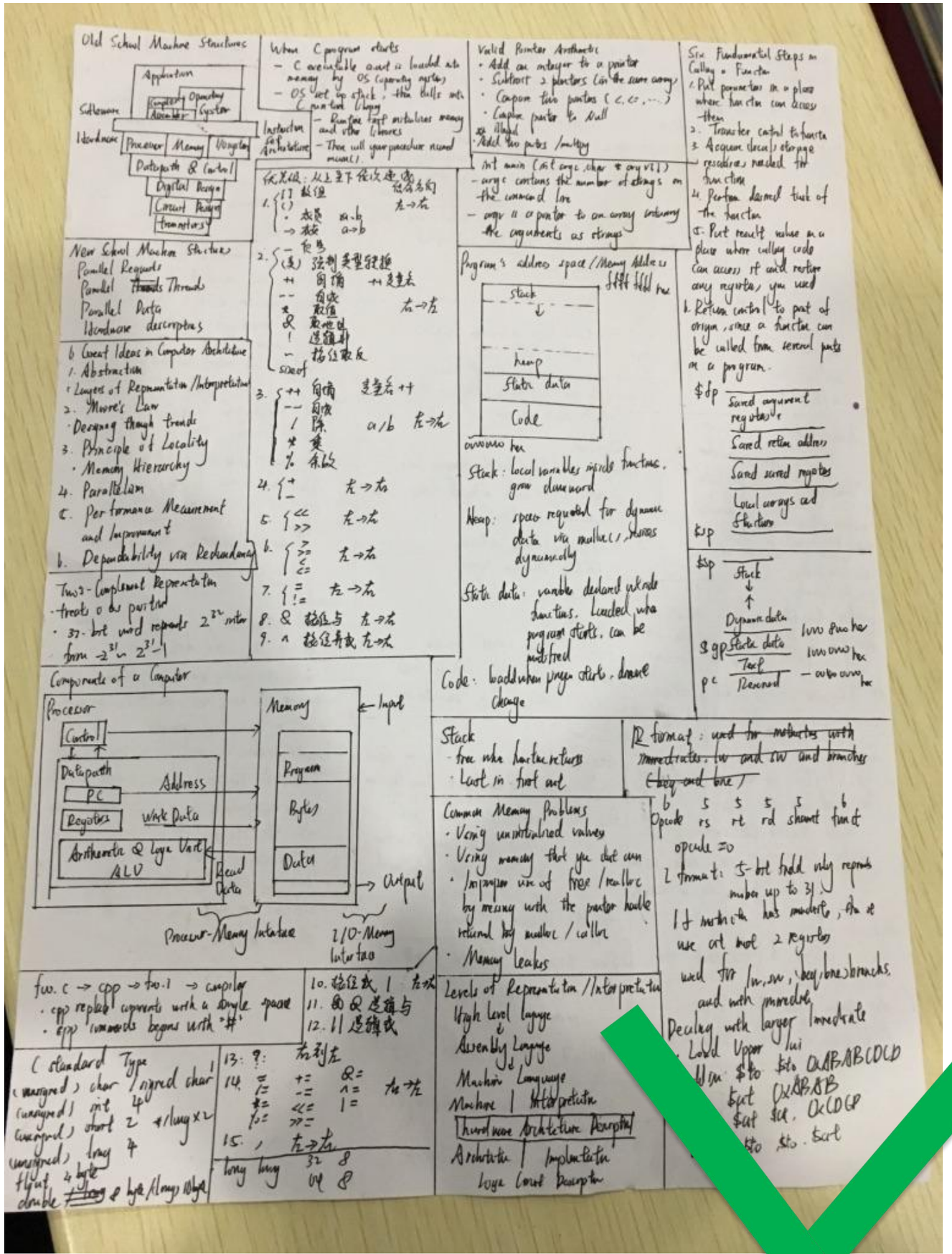

COMPUTER
ORGANIZATION
AND DESIGN
THE HA WARE/SOFTWA INTERFACE
RISC-V EDITION
DAVID A. PATTERSON
JOHN L. HENNESSY
MK
MORGAN KAUFMANN

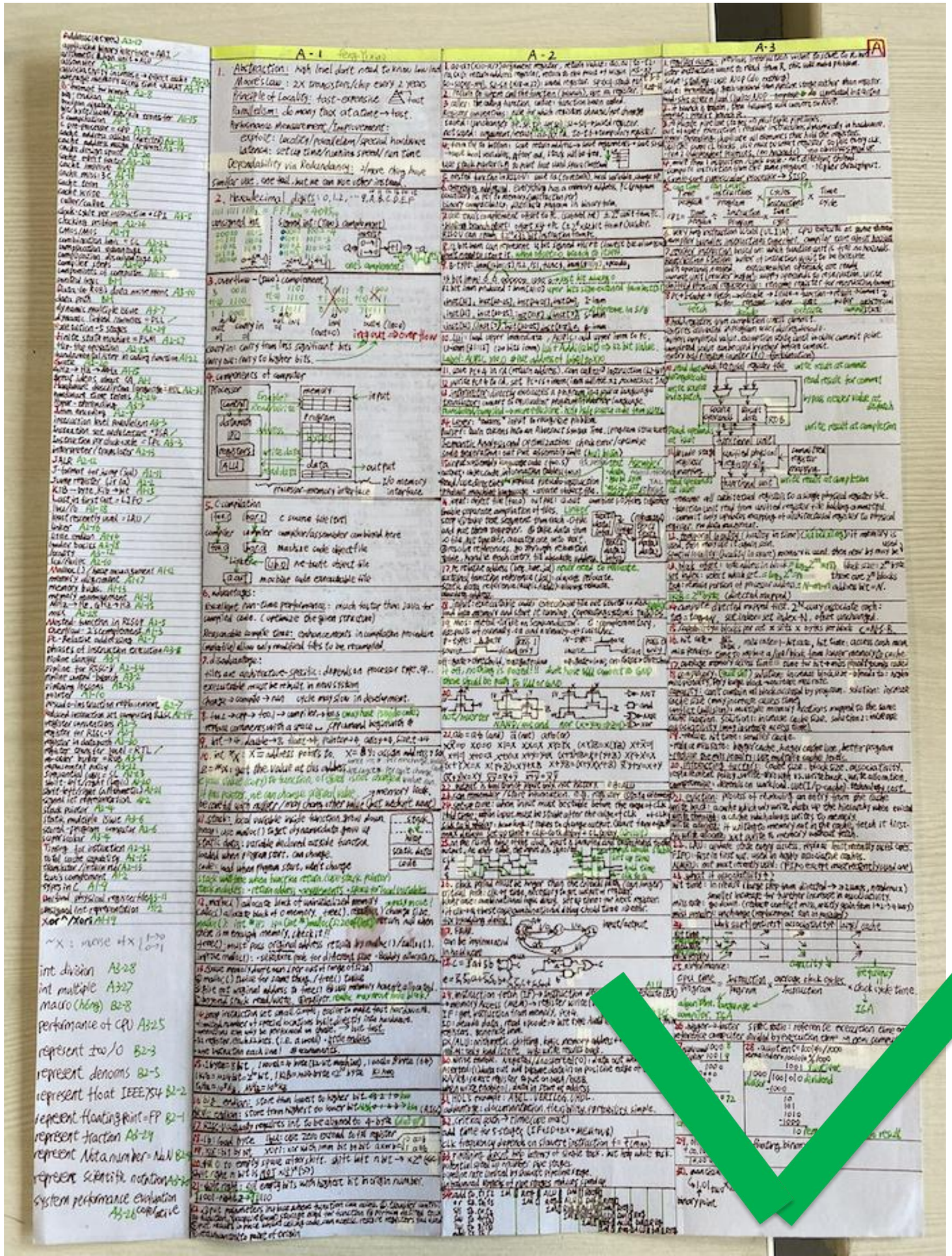
Cheat Sheet

- 1 A4 Cheat Sheet allowed (double sided)
  - Midterm II: 2 pages
  - Final: 3 pages
- Rules:
  - ***Hand-written*** – **not printed/photocopied!**
  - Your **name** in pinyin on the top!
  - Cheat Sheets not complying to this rule will be **confiscated**!


FUNCTIONS OF SEVERAL VARIABLES
LEVEL CURVES
FUNCTION OF n VARIABLES
PARTIAL DERIVATIVES
SECOND PARTIAL DERIVATIVES
CLAIRAUT'S THEOREM
PARTIAL DIFF. EQ'S
THE CHAIN RULE
IMPLICIT DIFFERENTIATION
TANGENT PLANE TO A LEVEL SURFACE
NORMAL LINE TO A LEVEL SURFACE
MAXIMUM AND MINIMUM VALUES
FINDING ABSOLUTE MAX. AND MINS.
MAXIMIZING AND MINIMIZING
DOMAINS: Allowed (x,y), (x,y,z) RANGES: 2's, 3's
ε-δ DEFINITION OF CONTINUITY
EQUATIONS OF TANGENT PLANES TO SURFACES
TOTAL DIFFERENTIAL
INCREMENTS Δx, Δy, Δz DIFFERENTIALS dx, dy, dz
THEOREM
DEPENDENCY DIAGRAMS
THE GRADIENT VECTOR
DIRECTIONAL DERIVATIVES
Add vector quantities head to tail!
NEWTON'S LAWS
x3.6
÷3.6
km/h


V = fλ = λ/T
LAW OF REFLECTION
REFRACTION IS THE BENDING OF LIGHT AROUND OBSTACLES AND THROUGH APERTURES
THE SUN IS AN
4H+ →


Old School Machine Structures
Software
Hardware
Application
Operating System
Compiler
Assembler
Processor
Memory
I/O system
Datapath & Control
Digital Design
Circuit Design
transistors
New School Machine Structures
Parallel Requests
Parallel Threads
Parallel Data
Hardware descriptions
6 Great Ideas in Computer Architecture
1. Abstraction
2. Moore's Law
3. Principle of Locality
4. Parallelism
5. Performance Measurement and Improvement
6. Dependability via Redundancy
When C program starts
Instruction Set Architecture
Valid Pointer Arithmetic
Six Fundamental Steps in Calling a Function
Program's address space / Memory Address
Stack
Heap
Static data
Code
Components of a Computer
Processor
Control
Datapath
PC
Registers
Arithmetic & Logic Unit ALU
Memory
Input
Output
Bytes
Processor-Memory Interface
I/O-Memory Interface
Stack
Common Memory Problems
Levels of Representation / Interpretation
High Level Language
Assembly Language
Machine Language
Hardware Architecture Description
Architecture
Implementation
Logic Circuit Description
R format
Opcode rs rt rd shamt funct
Standard Type
Decimal
Hexadecimal
Binary


A-1
A-2
A-3
1. Abstraction: high level don't need to know low level
Moore's Law: 2X transistors/chip every 2 years
Principle of Locality: fast-expensive
Parallelism: do many task at a time → fast.
Performance measurement/Improvement:
exploit: locality/parallelism/special hardware
latency: set up time/pipeline speed/ run time
Dependability via Redundancy: +more thing have
2. Hexadecimal digits: 0,1,2,...9,A,B,C,D,E,F
unsigned int
signed int: (2's) complement
3. overflow - (2's) complement
carry in
carry out
add (+ve)
sub (-ve)
int out → overflow
carry in: carry from less significant bits
carry out: carry to higher bits.
4. components of computer
Processor
memory
input
program
datapath
I/O
registers
ALU
data
output
I/O memory interface
processor-memory interface
5. C compilation
.c
.s
C source file (text)
compiler
compiler/assembler combined here
.o
.bin
machine code object file
linker
.exe
.out
machine code executable file
6. advantages:
Execution run-time performance: much faster than Java for compiled code. (optimize the given structure)
Responsible compile time: code errors/mistakes in compilation procedure (readable) allow early modifications to be revealed
7. disadvantage:
-files are architecture-specific: depends on processor type, OS...
executable must be rebuilt in new system
change → compile → run cycle may slow in development.
8. .c → .cpp → .ccc → .i → .s → .o → .exe (machine code)
.obj
.out
.dll
.lib
.a
.so
.ko
.elf
.dmp
.mod
.debug
.app
.pdb
.msi
.jar
.pyc
.war
.ear
.apk
.ipa
.nupkg
.whl
.egg
.gem
.snap
.appx
.msix
.xap
.ppa
.ppam
.reg
.dat
.bak
.tmp
.log
.cache
.swp
.lock
.pid
.core
.crash
.dump
.mdmp
.kdump
.vmcore
.hprof
.trace
.pstrace
.etl
.evtx
.journal
.wal
.shm
.db
.sqlite
.sqlite3
.realm
.ldb
.mdb
.accdb
.frm
.ibd
.myd
.myi
.ndf
.mdf
.ldf
.bak
.dmp
.exp
.imp
.dbf
.ora
.rdb
.fdb
.gdb
.nsf
.ntf
.dbx
.dbc
.dbs
.dbt
.dbv
.dbz
.db2
.db3
.db4
.db5
.db6
.db7
.db8
.db9
.db10
.db11
.db12
.db13
.db14
.db15
.db16
.db17
.db18
.db19
.db20
.db21
.db22
.db23
.db24
.db25
.db26
.db27
.db28
.db29
.db30
.db31
.db32
.db33
.db34
.db35
.db36
.db37
.db38
.db39
.db40
.db41
.db42
.db43
.db44
.db45
.db46
.db47
.db48
.db49
.db50
.db51
.db52
.db53
.db54
.db55
.db56
.db57
.db58
.db59
.db60
.db61
.db62
.db63
.db64
.db65
.db66
.db67
.db68
.db69
.db70
.db71
.db72
.db73
.db74
.db75
.db76
.db77
.db78
.db79
.db80
.db81
.db82
.db83
.db84
.db85
.db86
.db87
.db88
.db89
.db90
.db91
.db92
.db93
.db94
.db95
.db96
.db97
.db98
.db99
.db100
.db101
.db102
.db103
.db104
.db105
.db106
.db107
.db108
.db109
.db110
.db111
.db112
.db113
.db114
.db115
.db116
.db117
.db118
.db119
.db120
.db121
.db122
.db123
.db124
.db125
.db126
.db127
.db128
.db129
.db130
.db131
.db132
.db133
.db134
.db135
.db136
.db137
.db138
.db139
.db140
.db141
.db142
.db143
.db144
.db145
.db146
.db147
.db148
.db149
.db150
.db151
.db152
.db153
.db154
.db155
.db156
.db157
.db158
.db159
.db160
.db161
.db162
.db163
.db164
.db165
.db166
.db167
.db168
.db169
.db170
.db171
.db172
.db173
.db174
.db175
.db176
.db177
.db178
.db179
.db180
.db181
.db182
.db183
.db184
.db185
.db186
.db187
.db188
.db189
.db190
.db191
.db192
.db193
.db194
.db195
.db196
.db197
.db198
.db199
.db200
.db201
.db202
.db203
.db204
.db205
.db206
.db207
.db208
.db209
.db210
.db211
.db212
.db213
.db214
.db215
.db216
.db217
.db218
.db219
.db220
.db221
.db222
.db223
.db224
.db225
.db226
.db227
.db228
.db229
.db230
.db231
.db232
.db233
.db234
.db235
.db236
.db237
.db238
.db239
.db240
.db241
.db242
.db243
.db244
.db245
.db246
.db247
.db248
.db249
.db250
.db251
.db252
.db253
.db254
.db255
.db256
.db257
.db258
.db259
.db260
.db261
.db262
.db263
.db264
.db265
.db266
.db267
.db268
.db269
.db270
.db271
.db272
.db273
.db274
.db275
.db276
.db277
.db278
.db279
.db280
.db281
.db282
.db283
.db284
.db285
.db286
.db287
.db288
.db289
.db290
.db291
.db292
.db293
.db294
.db295
.db296
.db297
.db298
.db299
.db300
.db301
.db302
.db303
.db304
.db305
.db306
.db307
.db308
.db309
.db310
.db311
.db312
.db313
.db314
.db315
.db316
.db317
.db318
.db319
.db320
.db321
.db322
.db323
.db324
.db325
.db326
.db327
.db328
.db329
.db330
.db331
.db332
.db333
.db334
.db335
.db336
.db337
.db338
.db339
.db340
.db341
.db342
.db343
.db344
.db345
.db346
.db347
.db348
.db349
.db350
.db351
.db352
.db353
.db354
.db355
.db356
.db357
.db358
.db359
.db360
.db361
.db362
.db363
.db364
.db365
.db366
.db367
.db368
.db369
.db370
.db371
.db372
.db373
.db374
.db375
.db376
.db377
.db378
.db379
.db380
.db381
.db382
.db383
.db384
.db385
.db386
.db387
.db388
.db389
.db390
.db391
.db392
.db393
.db394
.db395
.db396
.db397
.db398
.db399
.db400
.db401
.db402
.db403
.db404
.db405
.db406
.db407
.db408
.db409
.db410
.db411
.db412
.db413
.db414
.db415
.db416
.db417
.db418
.db419
.db420
.db421
.db422
.db423
.db424
.db425
.db426
.db427
.db428
.db429
.db430
.db431
.db432
.db433
.db434
.db435
.db436
.db437
.db438
.db439
.db440
.db441
.db442
.db443
.db444
.db445
.db446
.db447
.db448
.db449
.db450
.db451
.db452
.db453
.db454
.db455
.db456
.db457
.db458
.db459
.db460
.db461
.db462
.db463
.db464
.db465
.db466
.db467
.db468
.db469
.db470
.db471
.db472
.db473
.db474
.db475
.db476
.db477
.db478
.db479
.db480
.db481
.db482
.db483
.db484
.db485
.db486
.db487
.db488
.db489
.db490
.db491
.db492
.db493
.db494
.db495
.db496
.db497
.db498
.db499
.db500
.db501
.db502
.db503
.db504
.db505
.db506
.db507
.db508
.db509
.db510
.db511
.db512
.db513
.db514
.db515
.db516
.db517
.db518
.db519
.db520
.db521
.db522
.db523
.db524
.db525
.db526
.db527
.db528
.db529
.db530
.db531
.db532
.db533
.db534
.db535
.db536
.db537
.db538
.db539
.db540
.db541
.db542
.db543
.db544
.db545
.db546
.db547
.db548
.db549
.db550
.db551
.db552
.db553
.db554
.db555
.db556
.db557
.db558
.db559
.db560
.db561
.db562
.db563
.db564
.db565
.db566
.db567
.db568
.db569
.db570
.db571
.db572
.db573
.db574
.db575
.db576
.db577
.db578
.db579
.db580
.db581
.db582
.db583
.db584
.db585
.db586
.db587
.db588
.db589
.db590
.db591
.db592
.db593
.db594
.db595
.db596
.db597
.db598
.db599
.db600
.db601
.db602
.db603
.db604
.db605
.db606
.db607
.db608
.db609
.db610
.db611
.db612
.db613
.db614
.db615
.db616
.db617
.db618
.db619
.db620
.db621
.db622
.db623
.db624
.db625
.db626
.db627
.db628
.db629
.db630
.db631
.db632
.db633
.db634
.db635
.db636
.db637
.db638
.db639
.db640
.db641
.db642
.db643
.db644
.db645
.db646
.db647
.db648
.db649
.db650
.db651
.db652
.db653
.db654
.db655
.db656
.db657
.db658
.db659
.db660
.db661
.db662
.db663
.db664
.db665
.db666
.db667
.db668
.db669
.db670
.db671
.db672
.db673
.db674
.db675
.db676
.db677
.db678
.db679
.db680
.db681
.db682
.db683
.db684
.db685
.db686
.db687
.db688
.db689
.db690
.db691
.db692
.db693
.db694
.db695
.db696
.db697
.db698
.db699
.db700
.db701
.db702
.db703
.db704
.db705
.db706
.db707
.db708
.db709
.db710
.db711
.db712
.db713
.db714
.db715
.db716
.db717
.db718
.db719
.db720
.db721
.db722
.db723
.db724
.db725
.db726
.db727
.db728
.db729
.db730
.db731
.db732
.db733
.db734
.db735
.db736
.db737
.db738
.db739
.db740
.db741
.db742
.db743
.db744
.db745
.db746
.db747
.db748
.db749
.db750
.db751
.db752
.db753
.db754
.db755
.db756
.db757
.db758
.db759
.db760
.db761
.db762
.db763
.db764
.db765
.db766
.db767
.db768
.db769
.db770
.db771
.db772
.db773
.db774
.db775
.db776
.db777
.db778
.db779
.db780
.db781
.db782
.db783
.db784
.db785
.db786
.db787
.db788
.db789
.db790
.db791
.db792
.db793
.db794
.db795
.db796
.db797
.db798
.db799
.db800
.db801
.db802
.db803
.db804
.db805
.db806
.db807
.db808
.db809
.db810
.db811
.db812
.db813
.db814
.db815
.db816
.db817
.db818
.db819
.db820
.db821
.db822
.db823
.db824
.db825
.db826
.db827
.db828
.db829
.db830
.db831
.db832
.db833
.db834
.db835
.db836
.db837
.db838
.db839
.db840
.db841
.db842
.db843
.db844
.db845
.db846
.db847
.db848
.db849
.db850
.db851
.db852
.db853
.db854
.db855
.db856
.db857
.db858
.db859
.db860
.db861
.db862
.db863
.db864
.db865
.db866
.db867
.db868
.db869
.db870
.db871
.db872
.db873
.db874
.db875
.db876
.db877
.db878
.db879
.db880
.db881
.db882
.db883
.db884
.db885
.db886
.db887
.db888
.db889
.db890
.db891
.db892
.db893
.db894
.db895
.db896
.db897
.db898
.db899
.db900
.db901
.db902
.db903
.db904
.db905
.db906
.db907
.db908
.db909
.db910
.db911
.db912
.db913
.db914
.db915
.db916
.db917
.db918
.db919
.db920
.db921
.db922
.db923
.db924
.db925
.db926
.db927
.db928
.db929
.db930
.db931
.db932
.db933
.db934
.db935
.db936
.db937
.db938
.db939
.db940
.db941
.db942
.db943
.db944
.db945
.db946
.db947
.db948
.db949
.db950
.db951
.db952
.db953
.db954
.db955
.db956
.db957
.db958
.db959
.db960
.db961
.db962
.db963
.db964
.db965
.db966
.db967
.db968
.db969
.db970
.db971
.db972
.db973
.db974
.db975
.db976
.db977
.db978
.db979
.db980
.db981
.db982
.db983
.db984
.db985
.db986
.db987
.db988
.db989
.db990
.db991
.db992
.db993
.db994
.db995
.db996
.db997
.db998
.db999
.db1000
7. disadvantage:
-files are architecture-specific: depends on processor type, OS...
executable must be rebuilt in new system
change → compile → run cycle may slow in development.
8. .c → .cpp → .ccc → .i → .s → .o → .exe (machine code)
.obj
.out
.dll
.lib
.a
.so
.ko
.elf
.dmp
.mod
.debug
.app
.pdb
.msi
.jar
.pyc
.war
.ear
.apk
.ipa
.nupkg
.whl
.egg
.gem
.snap
.appx
.msix
.xap
.ppa
.ppam
.reg
.dat
.bak
.tmp
.log
.cache
.swp
.lock
.pid
.core
.crash
.dump
.mdmp
.kdump
.vmcore
.hprof
.trace
.pstrace
.etl
.evtx
.journal
.wal
.shm
.db
.sqlite
.sqlite3
.realm
.ldb
.mdb
.accdb
.frm
.ibd
.myd
.myi
.ndf
.mdf
.ldf
.bak
.dmp
.exp
.imp
.dbf
.ora
.rdb
.fdb
.gdb
.nsf
.ntf
.dbx
.dbc
.dbs
.dbt
.dbv
.dbz
.db2
.db3
.db4
.db5
.db6
.db7
.db8
.db9
.db10
.db11
.db12
.db13
.db14
.db15
.db16
.db17
.db18
.db19
.db20
.db21
.db22
.db23
.db24
.db25
.db26
.db27
.db28
.db29
.db30
.db31
.db32
.db33
.db34
.db35
.db36
.db37
.db38
.db39
.db40
.db41
.db42
.db43
.db44
.db45
.db46
.db47
.db48
.db49
.db50
.db51
.db52
.db53
.db54
.db55
.db56
.db57
.db58
.db59
.db60
.db61
.db62
.db63
.db64
.db65
.db66
.db67
.db68
.db69
.db70
.db71
.db72
.db73
.db74
.db75
.db76
.db77
.db78
.db79
.db80
.db81
.db82
.db83
.db84
.db85
.db86
.db87
.db88
.db89
.db90
.db91
.db92
.db93
.db94
.db95
.db96
.db97
.db98
.db99
.db100
.db101
.db102
.db103
.db104
.db105
.db106
.db107
.db108
.db109
.db110
.db111
.db112
.db113
.db114
.db115
.db116
.db117
.db118
.db119
.db120
.db121
.db122
.db123
.db124
.db125
.db126
.db127
.db128
.db129
.db130
.db131
.db132
.db133
.db134
.db135
.db136
.db137
.db138
.db139
.db140
.db141
.db142
.db143
.db144
.db145
.db146
.db147
.db148
.db149
.db150
.db151
.db152
.db153
.db154
.db155
.db156
.db157
.db158
.db159
.db160
.db161
.db162
.db163
.db164
.db165
.db166
.db167
.db168
.db169
.db170
.db171
.db172
.db173
.db174
.db175
.db176
.db177
.db178
.db179
.db180
.db181
.db182
.db183
.db184
.db185
.db186
.db187
.db188
.db189
.db190
.db191
.db192
.db193
.db194
.db195
.db196
.db197
.db198
.db199
.db200
.db201
.db202
.db203
.db204
.db205
.db206
.db207
.db208
.db209
.db210
.db211
.db212
.db213
.db214
.db215
.db216
.db217
.db218
.db219
.db220
.db221
.db222
.db223
.db224
.db225
.db226
.db227
.db228
.db229
.db230
.db231
.db232
.db233
.db234
.db235
.db236
.db237
.db238
.db239
.db240
.db241
.db242
.db243
.db244
.db245
.db246
.db247
.db248
.db249
.db250
.db251
.db252
.db253
.db254
.db255
.db256
.db257
.db258
.db259
.db260
.db261
.db262
.db263
.db264
.db265
.db266
.db267
.db268
.db269
.db270
.db271
.db272
.db273
.db274
.db275
.db276
.db277
.db278
.db279
.db280
.db281
.db282
.db283
.db284
.db285
.db286
.db287
.db288
.db289
.db290
.db291
.db292
.db293
.db294
.db295
.db296
.db297
.db298
.db299
.db300
.db301
.db302
.db303
.db304
.db305
.db306
.db307
.db308
.db309
.db310
.db311
.db312
.db313
.db314
.db315
.db316
.db317
.db318
.db319
.db320
.db321
.db322
.db323
.db324
.db325
.db326
.db327
.db328
.db329
.db330
.db331
.db332
.db333
.db334
.db335
.db336
.db337
.db338
.db339
.db340
.db341
.db342
.db343
.db344
.db345
.db346
.db347
.db348
.db349
.db350
.db351
.db352
.db353
.db354
.db355
.db356
.db357
.db358
.db359
.db360
.db361
.db362
.db363
.db364
.db365
.db366
.db367
.db368
.db369
.db370
.db371
.db372
.db373
.db374
.db375
.db376
.db377
.db378
.db379
.db380
.db381
.db382
.db383
.db384
.db385
.db386
.db387
.db388
.db389
.db390
.db391
.db392
.db393
.db394
.db395
.db396
.db397
.db398
.db399
.db400
.db401
.db402
.db403
.db404
.db405
.db406
.db407
.db408
.db409
.db410
.db411
.db412
.db413
.db414
.db415
.db416
.db417
.db418
.db419
.db420
.db421
.db422
.db423
.db424
.db425
.db426
.db427
.db428
.db429
.db430
.db431
.db432
.db433
.db434
.db435
.db436
.db437
.db438
.db439
.db440
.db441
.db442
.db443
.db444
.db445
.db446
.db447
.db448
.db449
.db450
.db451
.db452
.db453
.db454
.db455
.db456
.db457
.db458
.db459
.db460
.db461
.db462
.db463
.db464
.db465
.db466
.db467
.db468
.db469
.db470
.db471
.db472
.db473
.db474
.db475
.db476
.db477
.db478
.db479
.db480
.db481
.db482
.db483
.db484
.db485
.db486
.db487
.db488
.db489
.db490
.db491
.db492
.db493
.db494
.db495
.db496
.db497
.db498
.db499
.db500
.db501
.db502
.db503
.db504
.db505
.db506
.db507
.db508
.db509
.db510
.db511
.db512
.db513
.db514
.db515
.db516
.db517
.db518
.db519
.db520
.db521
.db522
.db523
.db524
.db525
.db526
.db527
.db528
.db529
.db530
.db531
.db532
.db533
.db534
.db535
.db536
.db537
.db538
.db539
.db540
.db541
.db542
.db543
.db544
.db545
.db546
.db547
.db548
.db549
.db550
.db551
.db552
.db553
.db554
.db555
.db556
.db557
.db558
.db559
.db560
.db561
.db562
.db563
.db564
.db565
.db566
.db567
.db568
.db569
.db570
.db571
.db572
.db573
.db574
.db575
.db576
.db577
.db578
.db579
.db580
.db581
.db582
.db583
.db584
.db585
.db586
.db587
.db588
.db589
.db590
.db591
.db592
.db593
.db594
.db595
.db596
.db597
.db598
.db599
.db600
.db601
.db602
.db603
.db604
.db605
.db606
.db607
.db608
.db609
.db610
.db611
.db612
.db613
.db614
.db615
.db616
.db617
.db618
.db619
.db620
.db621
.db622
.db623
.db624
.db625
.db626
.db627
.db628
.db629
.db630
.db631
.db632
.db633
.db634
.db635
.db636
.db637
.db638
.db639
.db640
.db641
.db642
.db643
.db644
.db645
.db646
.db647
.db648
.db649
.db650
.db651
.db652
.db653
.db654
.db655
.db656
.db657
.db658
.db659
.db660
.db661
.db662
.db663
.db664
.db665
.db666
.db667
.db668
.db669
.db670
.db671
.db672
.db673
.db674
.db675
.db676
.db677
.db678
.db679
.db680
.db681
.db682
.db683
.db684
.db685
.db686
.db687
.db688
.db689
.db690
.db691
.db692
.db693
.db694
.db695
.db696
.db697
.db698
.db699
.db700
.db701
.db702
.db703
.db704
.db705
.db706
.db707
.db708
.db709
.db710
.db711
.db712
.db713
.db714
.db715
.db716
.db717
.db718
.db719
.db720
.db721
.db722
.db723
.db724
.db725
.db726
.db727
.db728
.db729
.db730
.db731
.db732
.db733
.db734
.db735
.db736
.db737
.db738
.db739
.db740
.db741
.db742
.db743
.db744
.db745
.db746
.db747
.db748
.db749
.db750
.db751
.db752
.db753
.db754
.db755
.db756
.db757
.db758
.db759
.db760
.db761
.db762
.db763
.db764
.db765
.db766
.db767
.db768
.db769
.db770
.db771
.db772
.db773
.db774
.db775
.db776
.db777
.db778
.db779
.db780
.db781
.db782
.db783
.db784
.db785
.db786
.db787
.db788
.db789
.db790
.db791
.db792
.db793
.db794
.db795
.db796
.db797
.db798
.db799
.db800
.db801
.db802
.db803
.db804
.db805
.db806
.db807
.db808
.db809
.db810
.db811
.db812
.db813
.db814
.db815
.db816
.db817
.db818
.db819
.db820
.db821
.db822
.db823
.db824
.db825
.db826
.db827
.db828
.db829
.db830
.db831
.db832
.db833
.db834
.db835
.db836
.db837
.db838
.db839
.db840
.db841
.db842
.db843
.db844
.db845
.db846
.db847
.db848
.db849
.db850
.db851
.db852
.db853
.db854
.db855
.db856
.db857
.db858
.db859
.db860
.db861
.db862
.db863
.db864
.db865
.db866
.db867
.db868
.db869
.db870
.db871
.db872
.db873
.db874
.db875
.db876
.db877
.db878
.db879
.db880
.db881
.db882
.db883
.db884
.db885
.db886
.db887
.db888
.db889
.db890
.db891
.db892
.db893
.db894
.db895
.db896
.db897
.db898
.db899
.db900
.db901
.db902
.db903
.db904
.db905
.db906
.db907
.db908
.db909
.db910
.db911
.db912
.db913
.db914
.db915
.db916
.db917
.db918
.db919
.db920
.db921
.db922
.db923
.db924
.db925
.db926
.db927
.db928
.db929
.db930
.db931
.db932
.db933
.db934
.db935
.db936
.db937
.db938
.db939
.db940
.db941
.db942
.db943
.db944
.db945
.db946
.db947
.db948
.db949
.db950
.db951
.db952
.db953
.db954
.db955
.db956
.db957
.db958
.db959
.db960
.db961
.db962
.db963
.db964
.db965
.db966
.db967
.db968
.db969
.db970
.db971
.db972
.db973
.db974
.db975
.db976
.db977
.db978
.db979
.db980
.db981
.db982
.db983
.db984
.db985
.db986
.db987
.db988
.db989
.db990
.db991
.db992
.db993
.db994
.db995
.db996
.db997
.db998
.db999
.db1000
int division A3-28
int multiple A3-27
macro (Hong) B2-8
performance of CPU A3-25
represent two's C B2-3
represent denorms B2-5
represent float IEEE754 B2-2
represent floating&int=FP B2-1
represent fraction A3-24
represent Not a number=NaN B2-4
represent scientific notation A3-23
system performance evaluation A3-28 comparative

SIFT REFERENCE GUIDE (V.1.1) – CREATING TIMELINES WITH THE SIFT WORKSTATION

SANS COMPUTER FORENSICS and INCIDENT RESPONSE

THE PURPOSE OF THIS REFERENCE GUIDE IS TO WALK THROUGH THE PROCESS OF BOOTING THE SIFT WORKSTATION, CREATING A TIMELINE ("SUPER" OR "MICRO") AND REVIEWING IT.

1. VISIT: http://computer-forensics11.sans.org/community/downloads

1

oad: SIFT Workstation VM Appliance

Do FT Workstation Installation

2

2. BOOT SIFT VM

Login: *sansforensics*
Password: *forensics*

3

3. ELEVATE PRIVS

$ sudo su

4

4. CONNECT IMAGE TO SIFT

Plug hard drive to physical host and attach to SIFT VM

5

log2timeline PARSING PLUGINS

apache2_error - Apache2 error log file
chrome - Chrome history file
encase_dirlisting - CSV file that is exported from encase
evt - Windows 2k/XP/2k3 Event Log
evtx - Windows Event Log File (EVTX)
exif - Metadata information from files using ExifTool
ff_bookmark - Firefox bookmark file
firefox2 - Firefox 2 browser history
firefox3 - Firefox 3 history file
ftk_dirlisting - CSV file that is exported from FTK Imager (dirlisting)
generic_linux - Generic Linux logs that start with MMM DD HH:MM:SS
iehistory - index.dat file containg IE history
iis - IIS W3C log file
isatxt - ISA text export log file
jp_ntfs_change - CSV output file from JP (NTFS Change log)
mactime - Body file in the mactime format
mcafee - Log file
mft - NTFS MFT file
mssql_errlog - ERRORLOG file produced by MS SQL server
ntuser - NTUSER.DAT registry file
opera - Opera's global history file
oxml - OpenXML document pcap
pcap - PCAP file
pdf - Available PDF document metadata
prefetch - Prefetch directory
recycler - Recycle bin directory
restore 0.9 - Restore point directory
safari - Safari History.plist file
sam - SAM registry file
security - SECURITY registry file
setupapi - SetupAPI log file in Windows XP
skype_sql - Skype database
software - SOFTWARE registry file
sol - .sol (LSO) or a Flash cookie file
squid - Squid access log (http_emulate off)
syslog - Linux Syslog log file
system - SYSTEM registry file
tln - Body file in the TLN format
volatility - Volatility output files (psscan2, sockscan2, ...)
win_link - Windows shortcut file (or a link file)
wmiprov - wmiprov log file
xpfirewall - XP Firewall log

List plugins # log2timeline -f list
...HELP EXPAND THIS LIST. BUILD PLUGINS!!!

BY DAVID NIDES (12/16/2011)
TWITTER: @DAVNADS
BLOG: DAVNADS.BLOGSPOT.COM
EMAIL: DNIDES@KPMG.COM
CREDITS TO: ED GOINGS, ROB LE... KRISTINN GUDJONSSON, KPMG &...!!
QUESTIONS/FEEDBACK–CONTACT US!

KEY
Red text – image/source
Blue text – mount point
Purple text – output file
Green text – log2timeline plugins
Brown text - TimeZone

5. HARD DRIVE MOUNTING *(if you are using log2timeline-sift and Single DD you can skip to 7-A)*

EWF/...

SINGLE OR SPLIT IMAGE (2 options):

...nt_ewf.py image.E01 /mnt/ewf

#e...mage.E01 /mnt/ewf/

Not Needed For 7-A

 # mount -t ntfs -o ro,loop,show_sys_files,streams_interface=wind... offset=#### /mnt/ewf/<image> /mnt/windows_mount/

MOUNT TO MOUNT POINT

DD

SINGLE IM...

 # mount -t ntfs -o ...w_sys_files,streams_interface=windows,offset=#### image.dd /m...ws_mount/

SPLIT IMAGE (2 step proc...

 # affuse image.001 /mnt/aff
 # mount –t ntfs-3g –o loop,ro,show_...nt/aff/<image> /mnt/windows_...

6

HOW TO CALCULATE THE OFFSET FOR MOUNTING

1. Run mmls to query partition layout
 # mmls image.E01
2. Identify partition and byte offset
3. (Partition byte offset) x (bytes per sector) = **offset #### to use!**
Example: 63 X 512 = 32256

Note: If needed, repeat for each partition. Make new mount point:
***# mkdir** /mnt/windows_mount2/*

6. log2timeline default timezone is set to examiner local host. To change use -z [TIMEZONE] option. To list all available timezones:
 # log2timeline -z list

7-A & 7-B

8

7-A: AUTOMATED SUPER TIMELINE CREATION

log2timeline-sift -o –z [TIMEZONE] -p [PARTITION #] -i [IMAGE FILE]

DISK IMAGE (prompt for partition, mount, and run):

XP **# log2timeline-sift –z EST5EDT -i image**

WIN7 **# log2timeline-sift -win7 -z EST5EDT -i image**

FOR PARTITION (mount and run using all applicable pl...

XP **# log2timeline-sift –z EST5EDT -p 0 -i partiti...**

WIN7 **# log2timeline-sift -win7 -z EST5EDT -...tion**

OTHER USAGE EXAMPLES:

Display list of available plu...
 # log2timeline -f list
Run log2timeline use... use only specific plugins:
 # log2timeline-sift ...refetch –z EST5EDT -i image.dd
Help (man page...
 # log2timeli...

8. CSV F...PUT (*/cases/timeline-output-folder*)

-dat...e event, in the format of MM/DD/YYYY
-t... of day, expressed in a 24h format, HH:MM:SS
-z...: the timezone that was used to call the tool with.
-...: MACB meaning of the fields, comp w/ mactime format.
-...rce: Source short name (i.e. registry entries are REG)
-sourcetype: Desc of the source ("Internet Explorer" instead of WEBHIST)
-type: Timestamp type (i.e. "Last Accessed", "Last Written")
-user: Username associated with the entry, if one is available.
-host: Hostname associated with the entry, if one is available.
-short: Contains less text than the full description field.
-desc: where majority info is stored, the actual parsed desc of the entry.
-version: Version number of the timestamp object.
-filename: Filename with the full path that contained the entry
-inode: inode number of the file being parsed.
-notes: Some input modules insert additional information in the form of a note, which comes here. Or it can be used during the review.
-format: Input module name used to parse the file.
-extra: Additional information parsed is joined together and put here.

...ANUAL "MICRO" TIMELINE CREATION

...TIONS] [-f FORMAT] [-z TIMEZONE] [-o OUTPUT MODULE] [-w ...G_FILE/LOG_DIR [--] [FORMAT FILE OPTIONS]

...EM METADATA (using log2timeline or fls)

...stem data w/log2timeline from mounted file system:
 # log... ...f mft -o mactime –r -z EST5EDT -w mft.bo...lume/
OR Extract M... ...age using Sleuthkit:
 # fls -m "" -o offs... ...dd > fls.body
Convert body file for... ...format w/ mactime:
 # mactime –b fls.body –d

ARTIFACTS (run l2l on mounted file s... ...h plugins recursively)

Extract artifacts w/ log2timeline and ru... ...d file system:
 # log2timeline -f firefox3,chrome -o mactim... ...EDT -w web.body /mnt/volume/
Convert body file format to CSV format w/ mactime:
 # mactime –b log2timeline.body –d > log2timeline.csv

HELP? OPTIONS? USAGE?
log2timeline -help
Log2timeline-sift -help
L2t_process -help

OTHER log2timeline OUTPUT FORMATS
Note: CSV is Default Output
-BeeDocs - Mac OS X visualization tool
-CEF - Common Event Format - ArcSight
-CFTL - XML file- CyberForensics TimeLab visualization tool
-CSV - comma separated value file
-Mactime - Both older and newer version of the format supported for use by TSK's mactime
-SIMILE - XML file - SIMILE timeline visualization widget
-SQLite - SQLite database
-TLN - Tab Delimited File
-TLN - Format used by some of H Carvey tools, expressed as a ASCII output
-TLNX - Format used by some of H Carvey tools, expressed as a XML document

9

9. FILTER TIMELINE

Filter timeline with date range to include only:
l2t_process -b timeline.csv MM-DD-YYYY..MM-DD-YYYY > filtered.csv
Filter timeline with keyword list (one term per line in keywords.txt):
l2t_process -b timeline.csv -k keywords.txt > filtered.csv
What sources are in your timeline?
awk–F , '{print $6;}' timeline.csv| grep–v sourcetype|sort | uniq
Find all LNK files that reference E Drive
grep"Shortcut LNK" timeline.csv| grep"E:"
FiindMountPoints2 entries that reference E Drive
grep"MountPoints2 key" timeline.csv} | grep"E drive"
grepUSB timeline.csv| grep"SetupAPILog"

File System	M	A	C	B
Ext2/3	Modified	Accessed	Changed	N/A
FAT	Written	Accessed	N/A	Created
NTFS	File Modified	Accessed	MFT Modified	Created
UFS	Modified	Accessed	Changed	N/A

10

10. CONNECT TO SIFT

✓ ...> SETTINGS -> OPTIONS -> Shared Fo...ers -> Always Enabled (Check)

✓ 2. SIFT Desktop > VMware-Shared-Drive

✓ Access from a Win Machine \\SIFTWORKSTATION

11

11. REVIEW TIMELINE

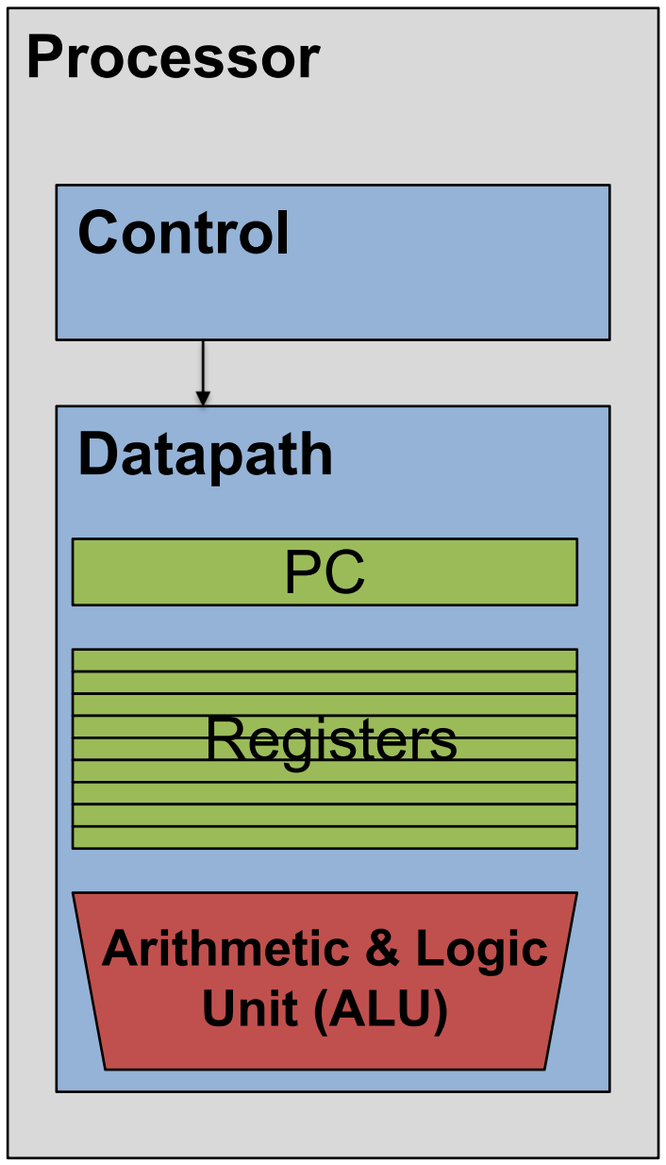
Review timelines using:
- Open, Soft, Filter with Excel
- Import into SPLUNK
- SIMILE
- Tapestry

Outline

- Datapath
  - Add building blocks and necessary components for different types of instructions, one type at a time
- Design of the controller
- Timing analysis

Controller & Datapath

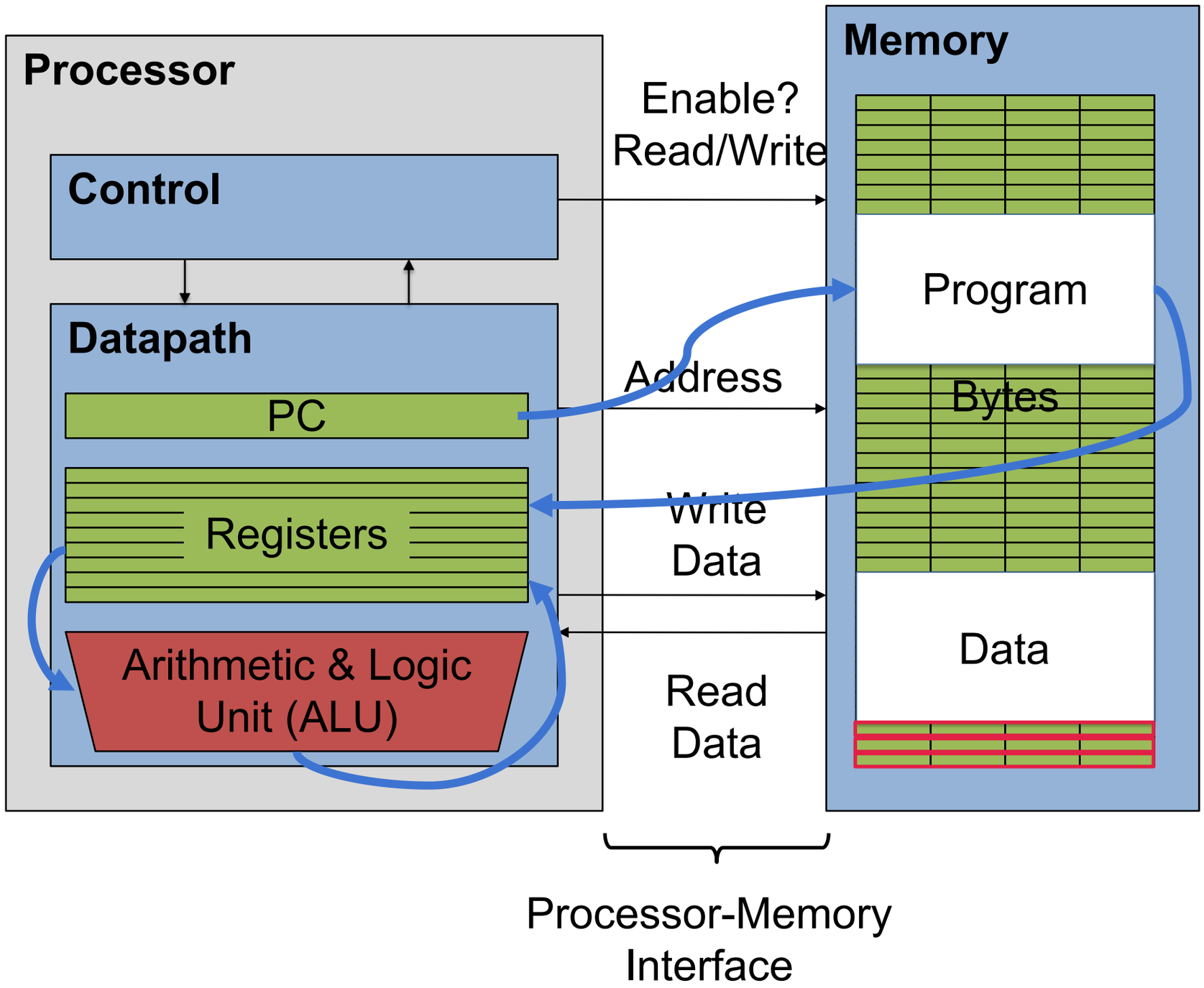
- A CPU that support RV32I can have so many states



- Consider the 32 registers alone
  - x0 always 0
  - Each bit in the other registers can be 0 or 1
- Not practical to enumerate all the state transitions
- Top-down design: build small modules and then connect them as needed
- Most digital systems can be divided into datapth and controller
  - Datapath contains data processing and storage
  - Controller controls data flow and state change (still can be modeled as FSM)
- Recall the execution of an instruction

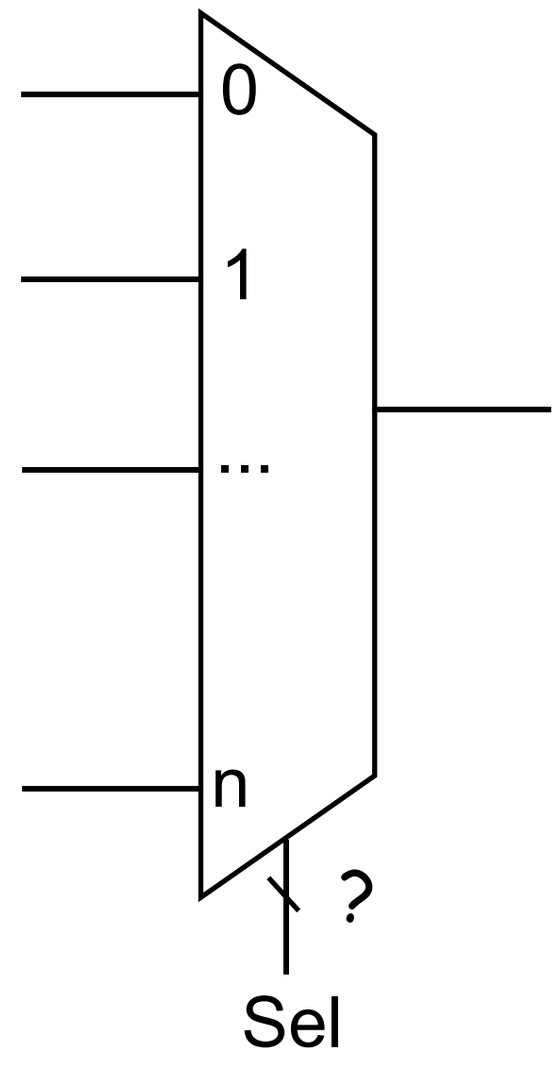
- Our Goal: Implement a RISC-V processor as a synchronous digital system (SDS).
- Each RV32I instruction can be done within 1 clock cycle (single-cycle CPU).

Datapath


Processor
Control
Datapath
PC
Registers
Arithmetic & Logic Unit (ALU)
Memory
Program
Bytes
Data
Enable?
Read/Write
Address
Write Data
Read Data
Processor-Memory Interface

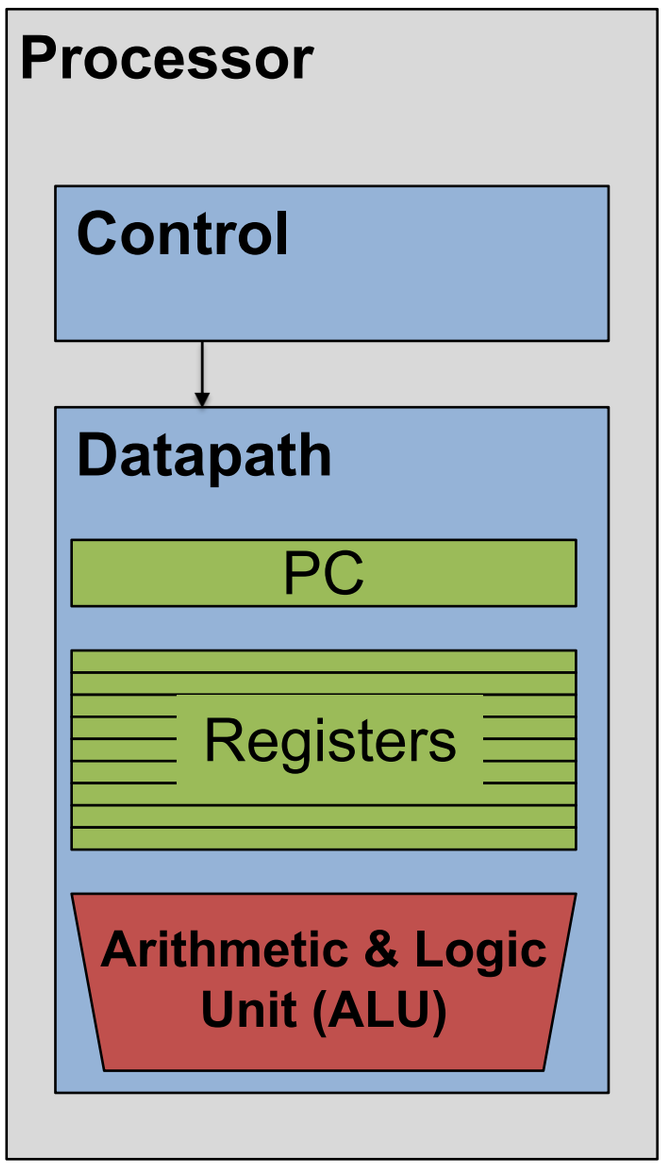
Multiplexer

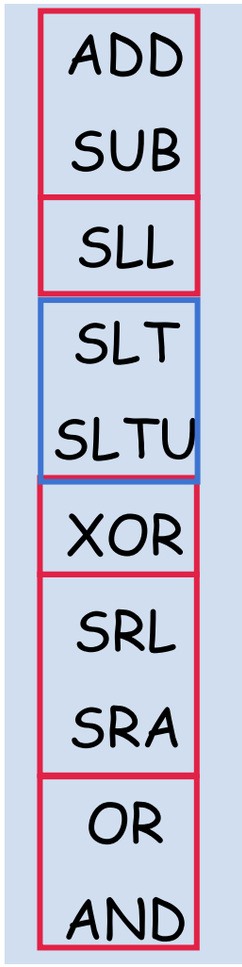
- N-to-1 multiplexer symbol

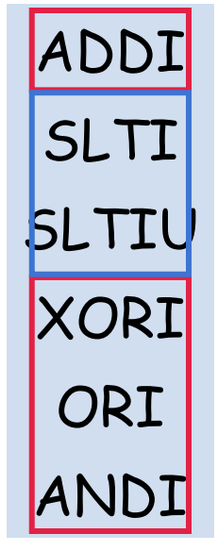


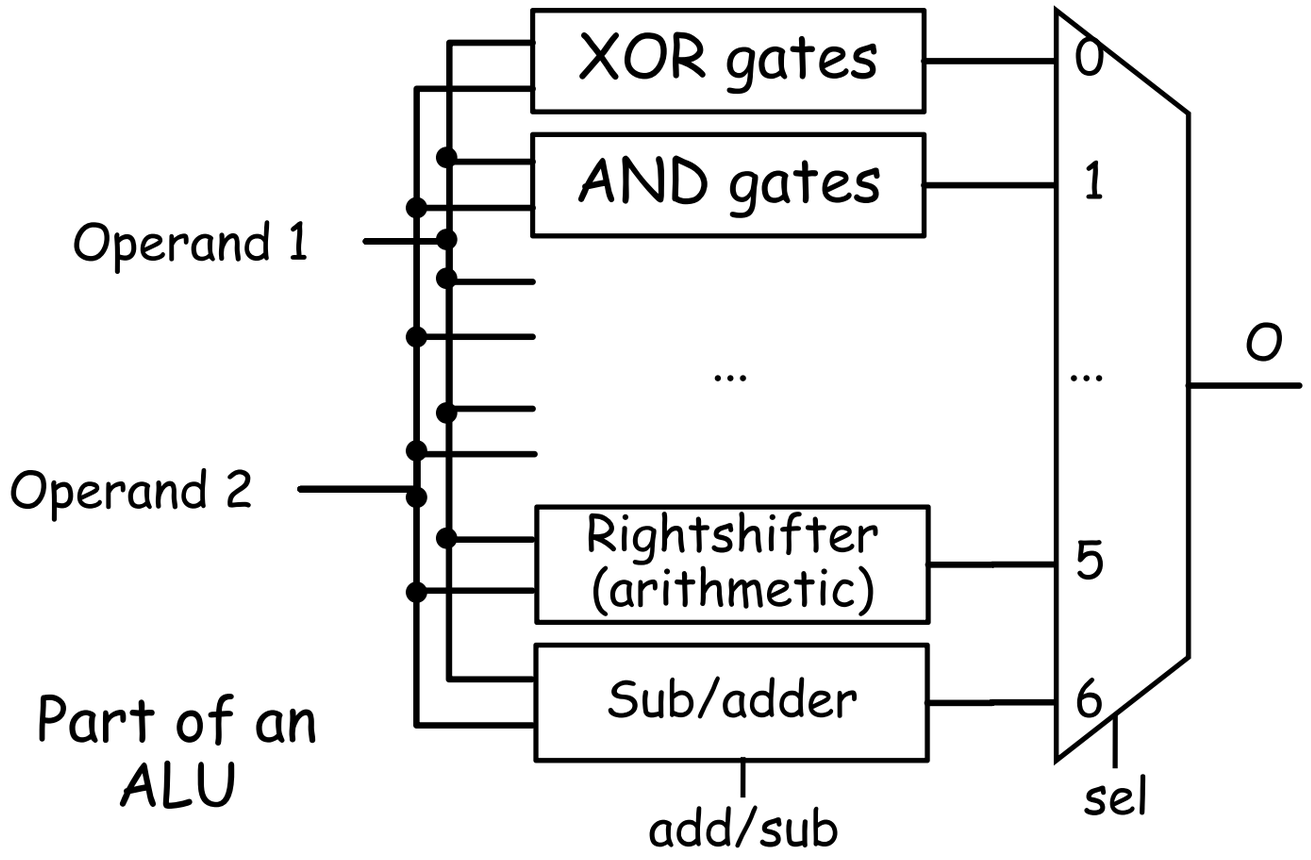
ALU

- An ALU should be able to execute all the arithmetic and logic operations







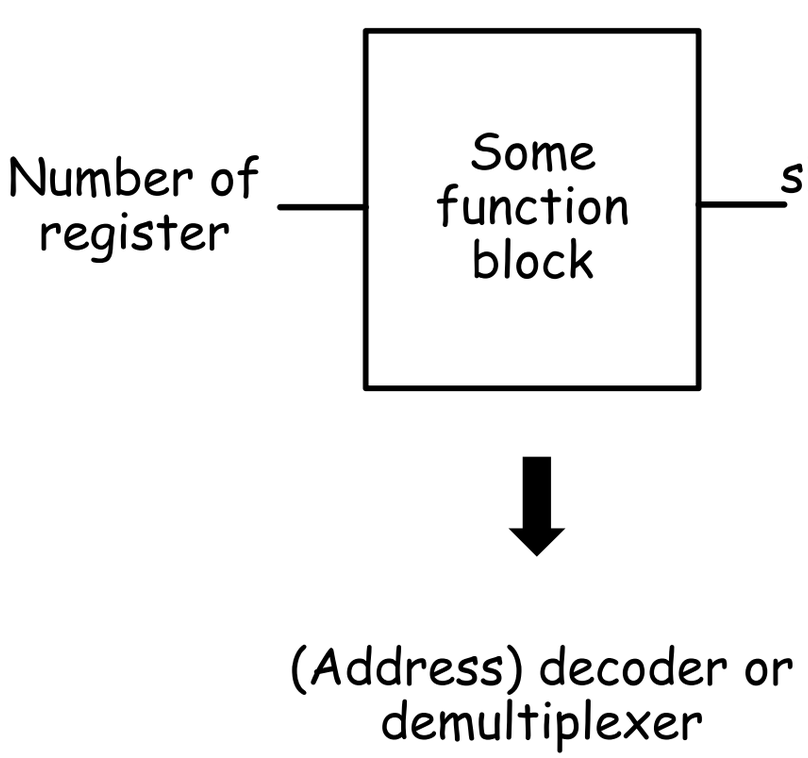


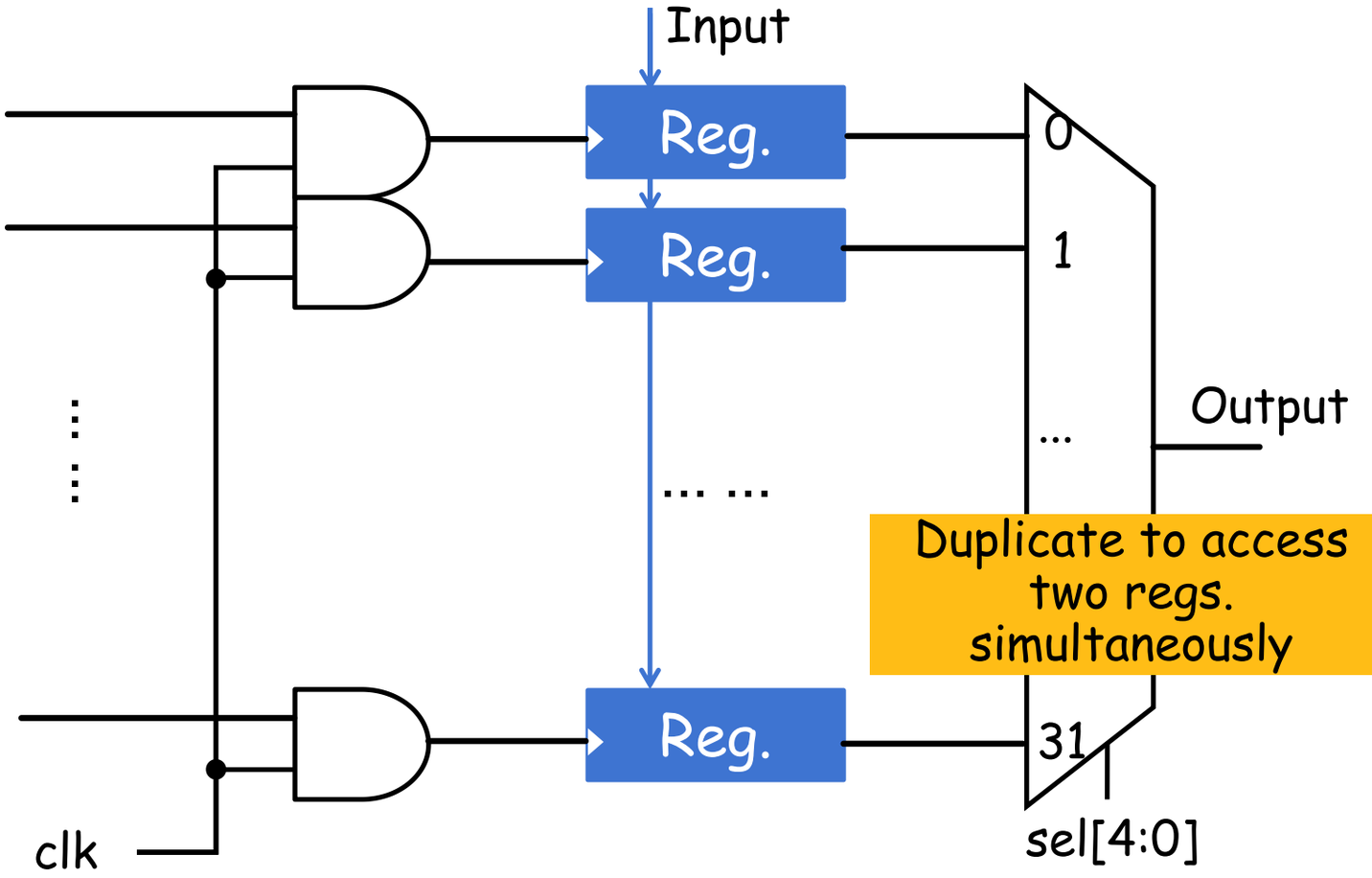
Note that all the signals expect the selection signals are 32-bit.

- ALU design that supports R-/I-arithmetic and logic operations completed

Register file

- The register file is the component that contains all the general purpose registers of the microprocessor
- A register file should provide data given the register numbers
- A register file should be able to change the stored value
  - How do we change values of a specific reg.?

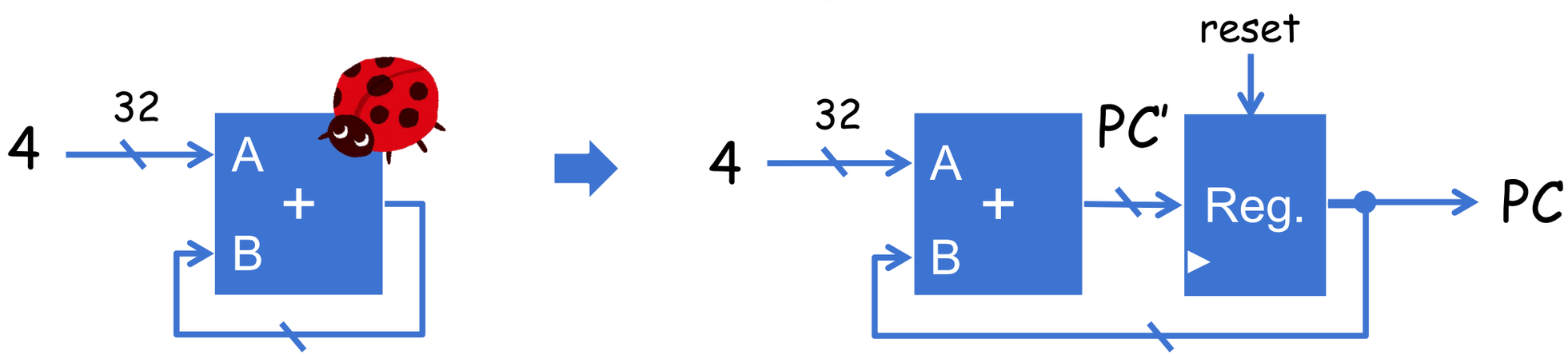




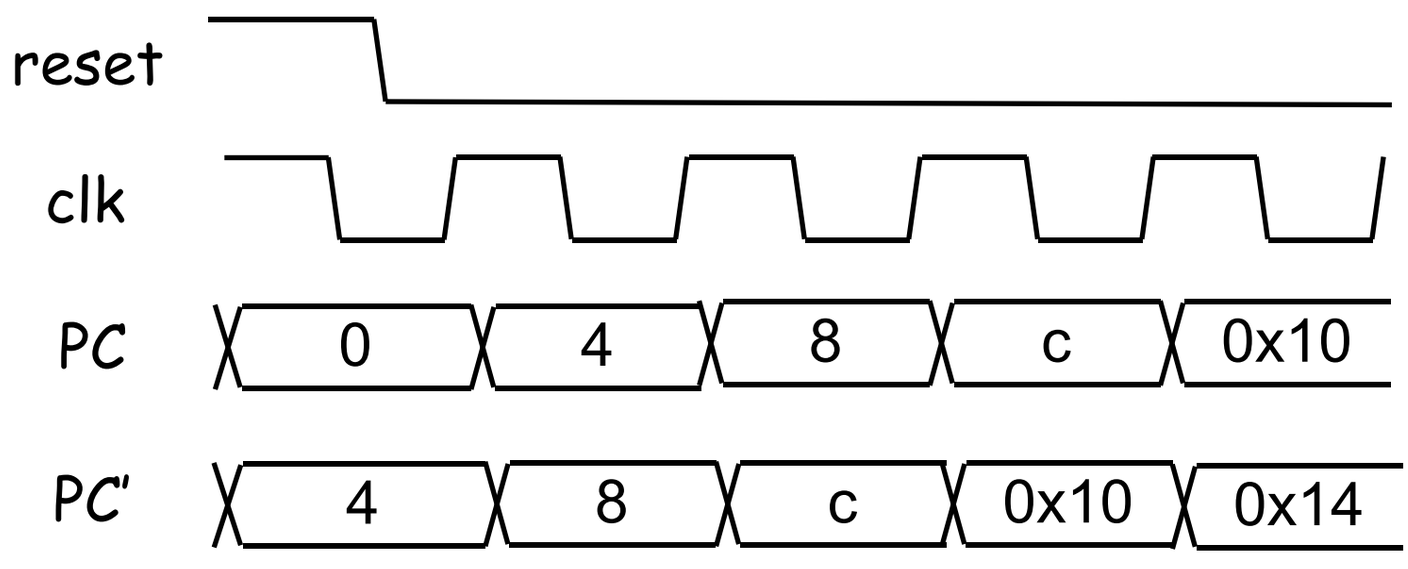
- Reg. file design completed

We have covered PC register previously

- Synchronous digital circuit can have feedback, e.g., iterative accumulator
  - e.g. PC = PC + 4 without considering branch or jump

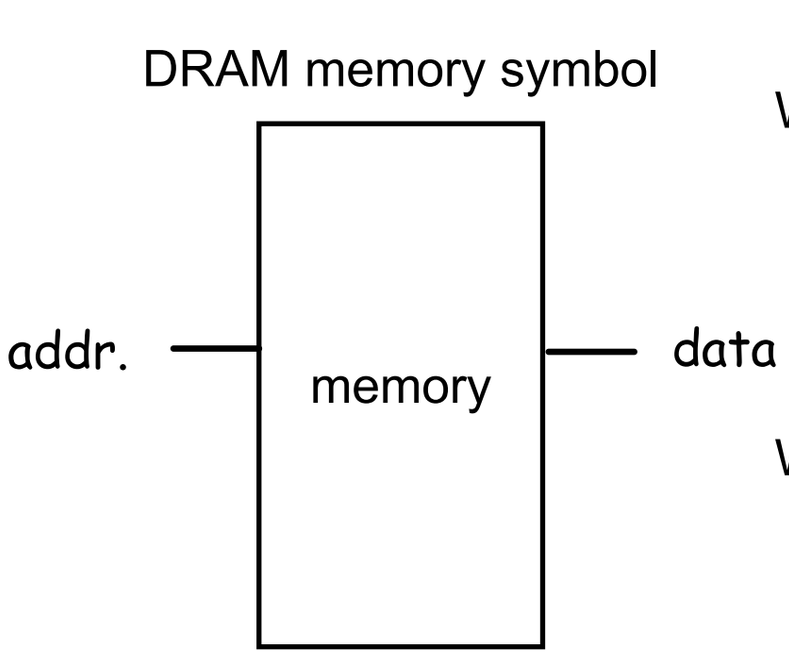


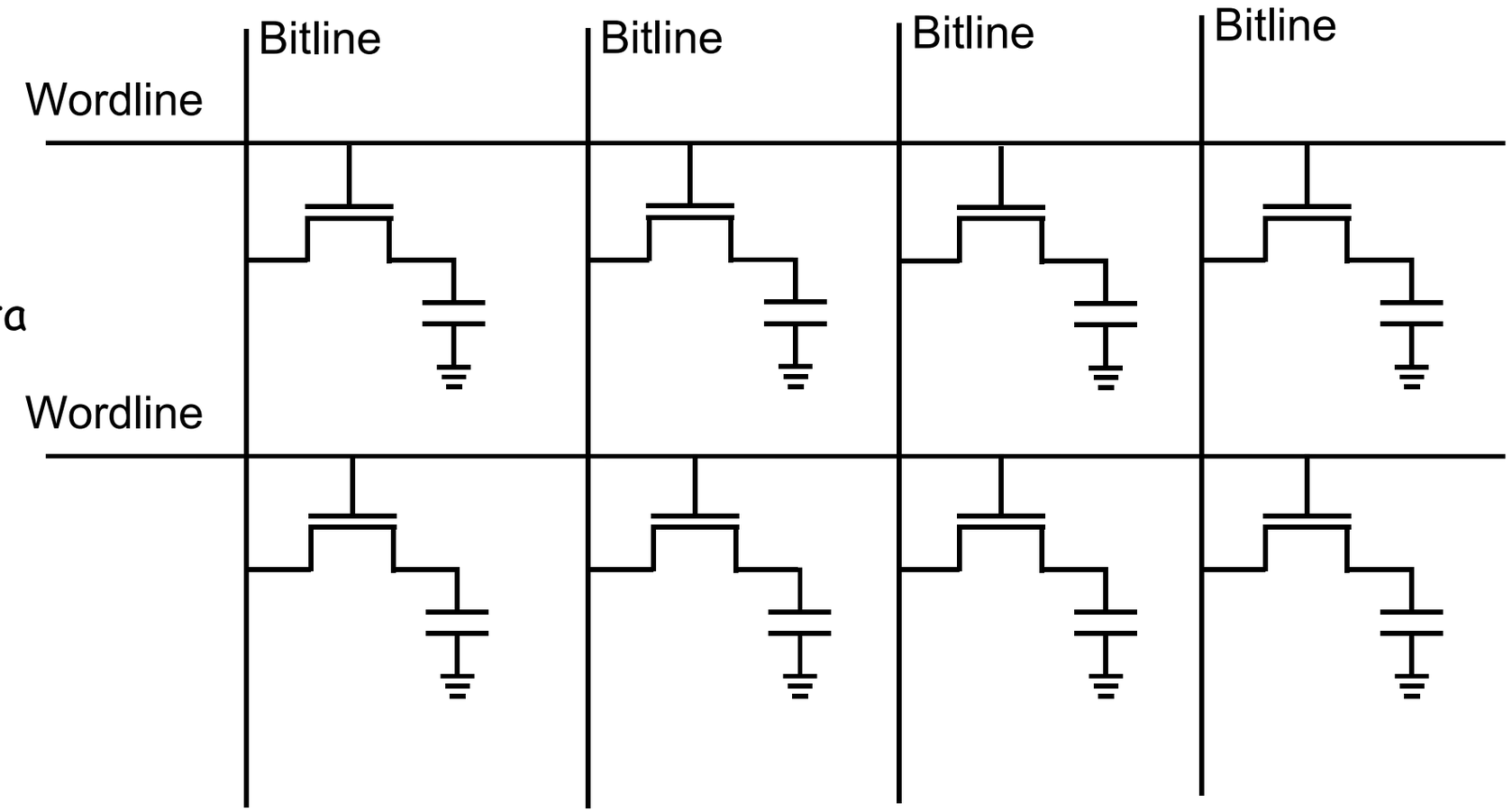
- Timing diagram



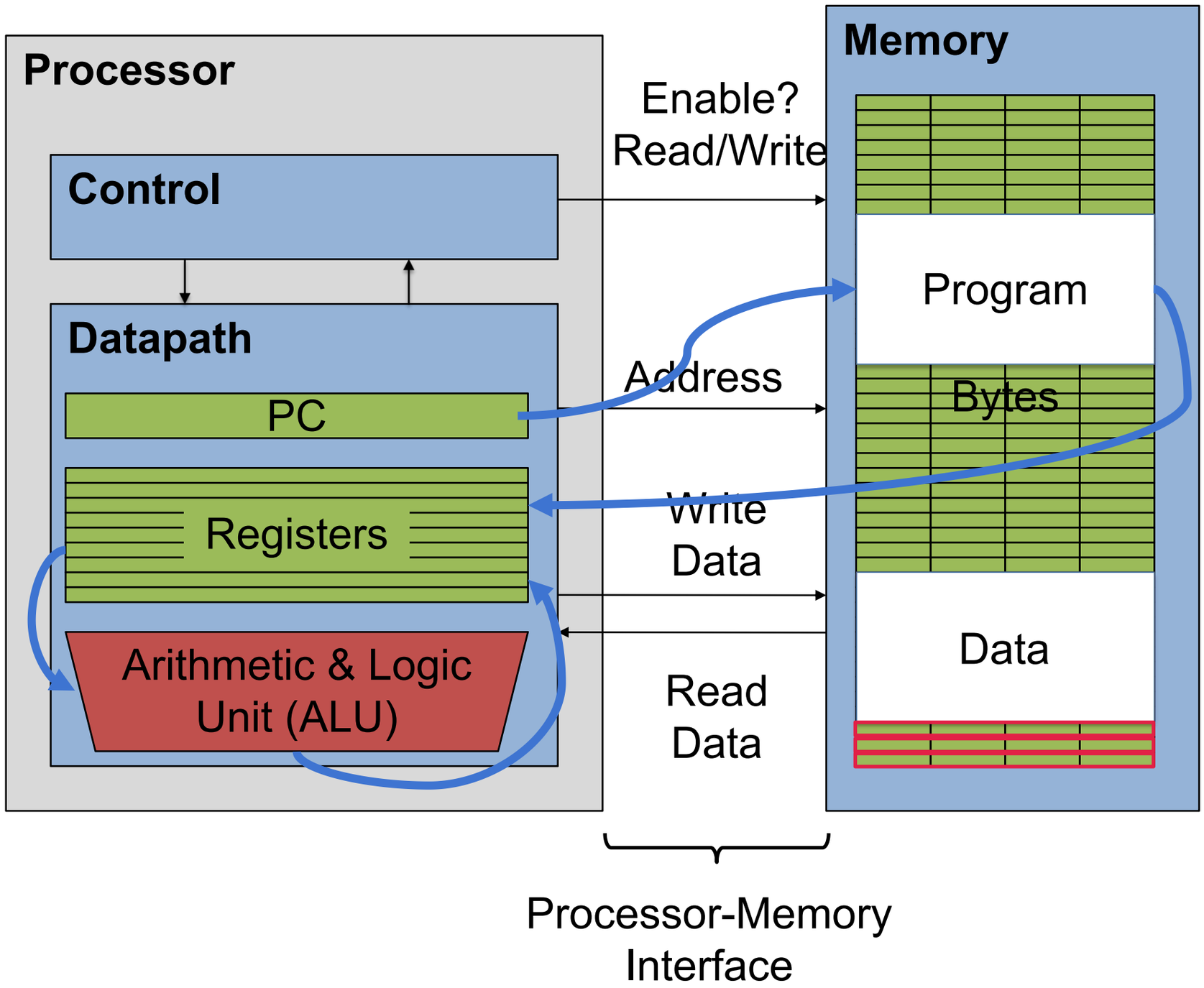
Useful building blocks-Memory

- Memory similar to register file except that the basic cell design is different
- Requires refresh for DRAM
- For ease of implementation, we only use its behavior model

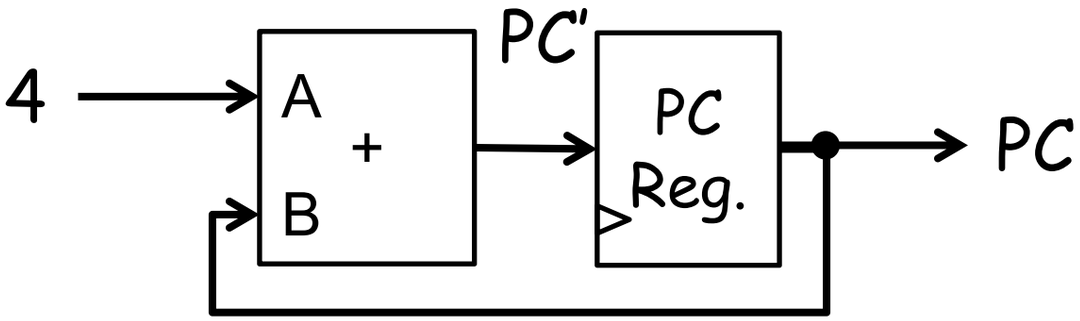




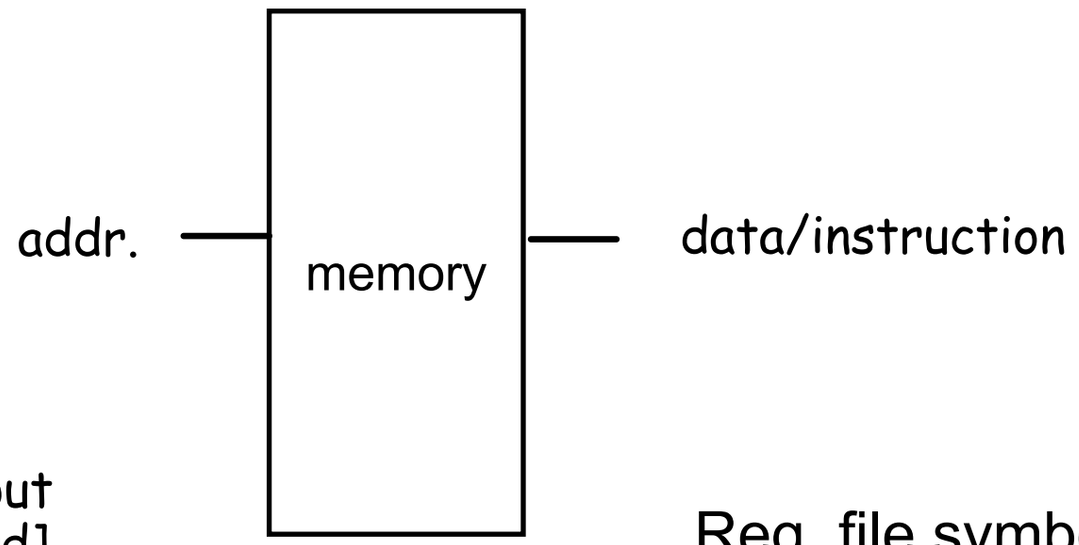
Datapath

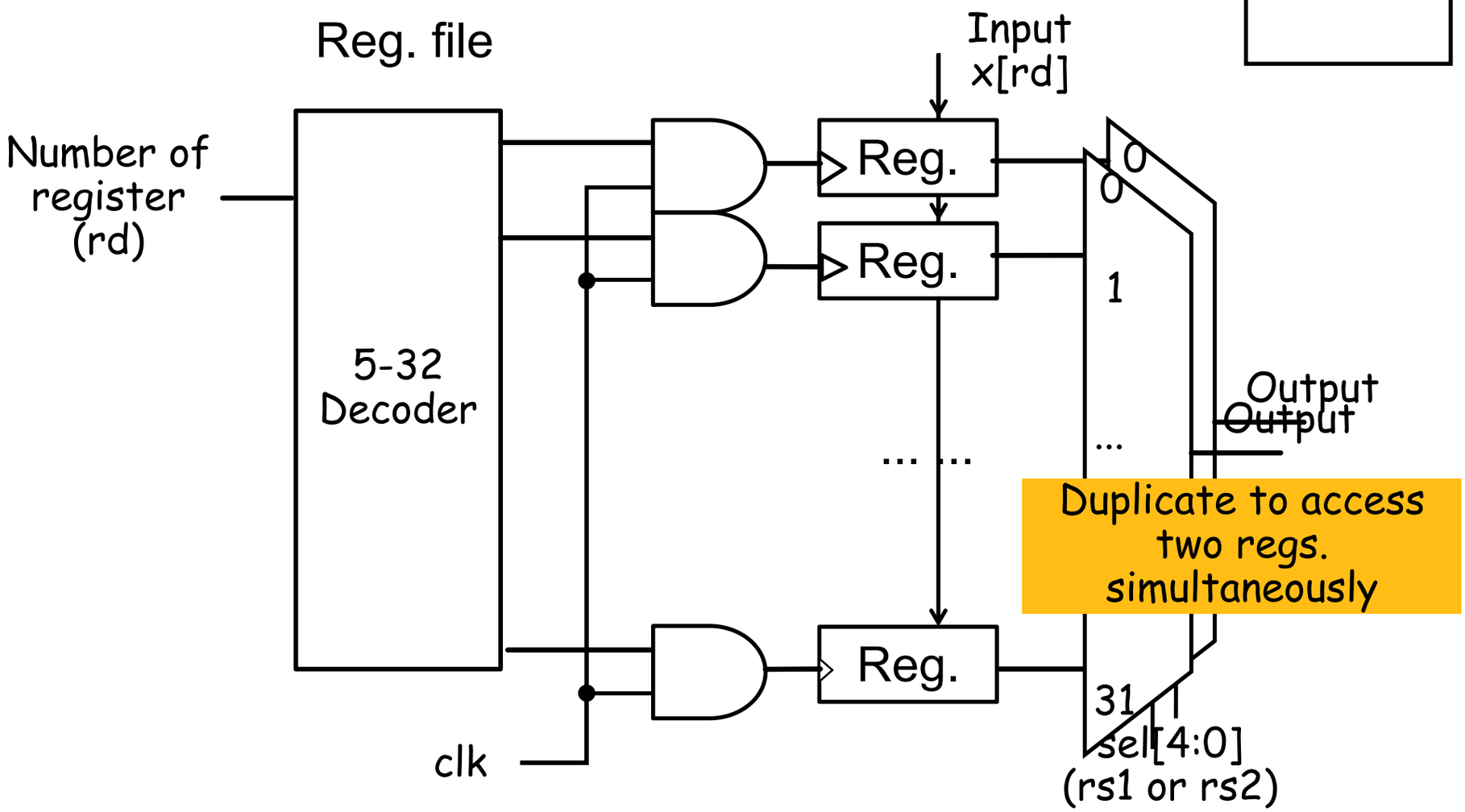

Processor
Control
Datapath
PC
Registers
Arithmetic & Logic Unit (ALU)
Enable?
Read/Write
Address
Write Data
Read Data
Memory
Program
Bytes
Data
Processor-Memory Interface

Datapath

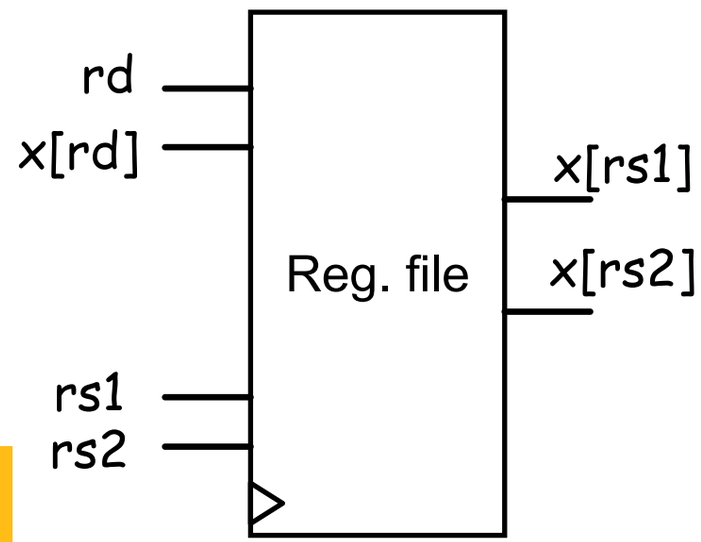


DRAM memory symbol



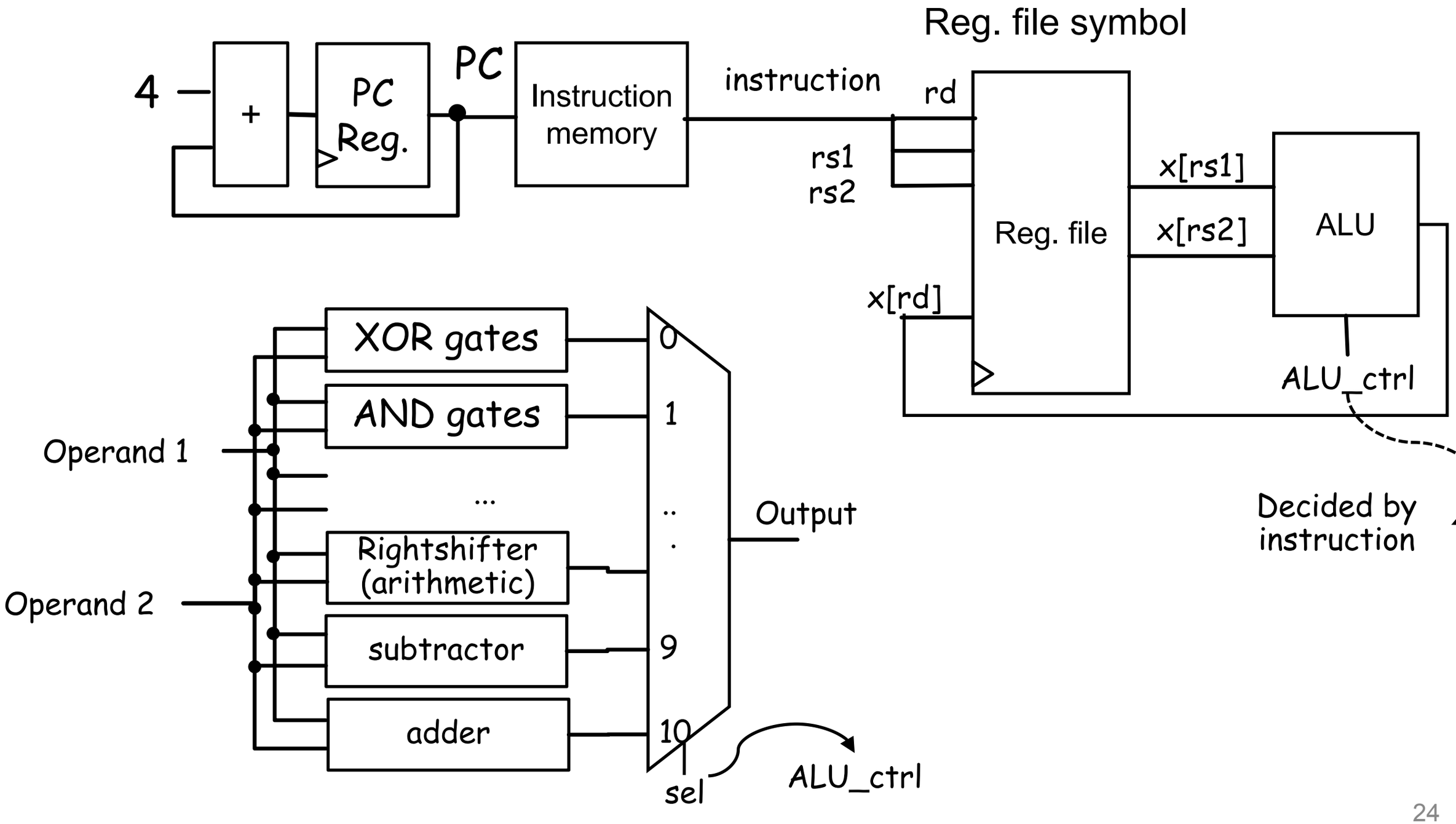


Reg. file symbol



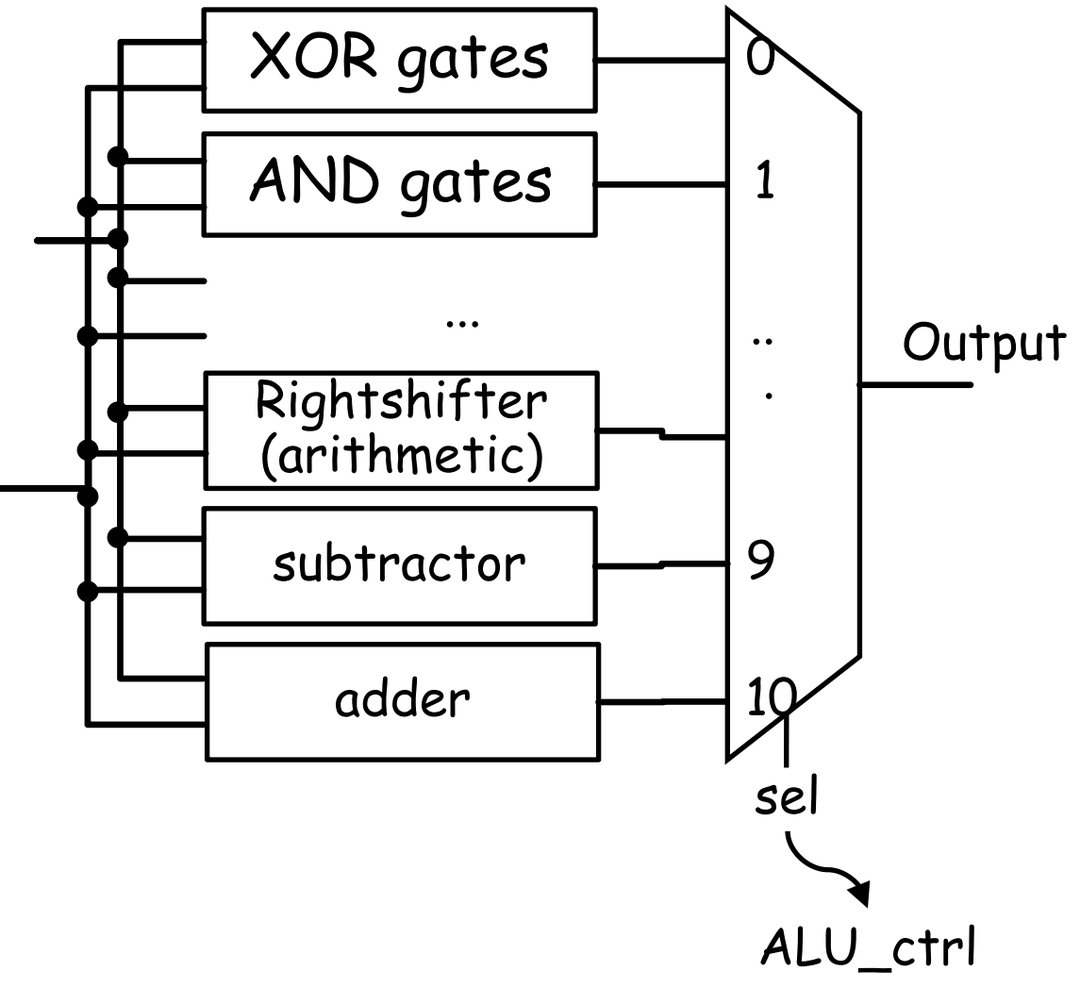
Datapath for R-type

- We have all the building blocks to execute R-type instructions



Datapath for R-type

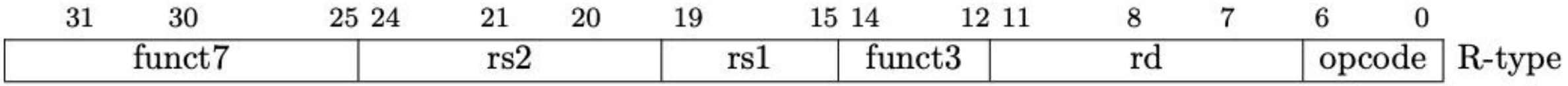
- We have all the building blocks to execute R-type instructions

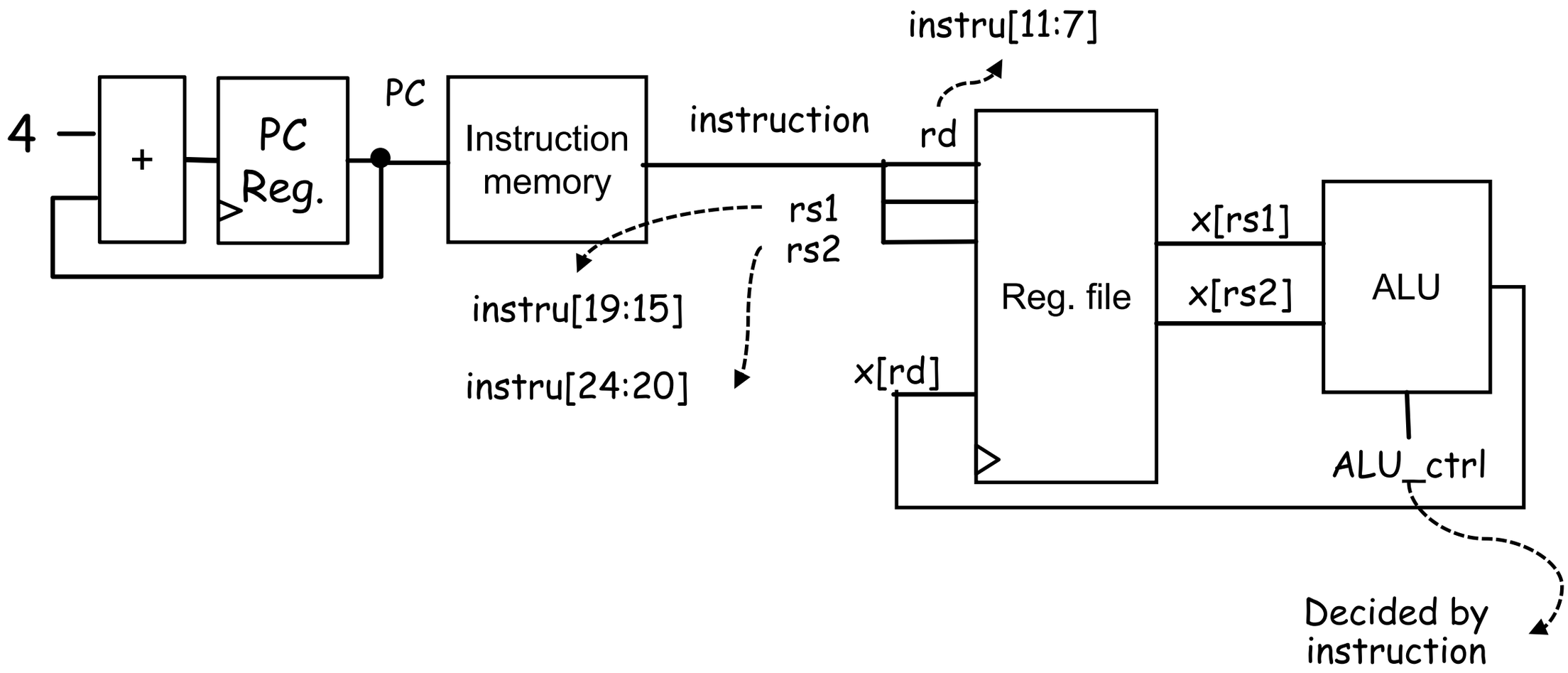


0000000	rs2	rs1	000	rd	0110011	ADD
0100000	rs2	rs1	000	rd	0110011	SUB
0000000	rs2	rs1	001	rd	0110011	SLL
0000000	rs2	rs1	010	rd	0110011	SLT
0000000	rs2	rs1	011	rd	0110011	SLTU
0000000	rs2	rs1	100	rd	0110011	XOR
0000000	rs2	rs1	101	rd	0110011	SRL
0100000	rs2	rs1	101	rd	0110011	SRA
0000000	rs2	rs1	110	rd	0110011	OR
0000000	rs2	rs1	111	rd	0110011	AND

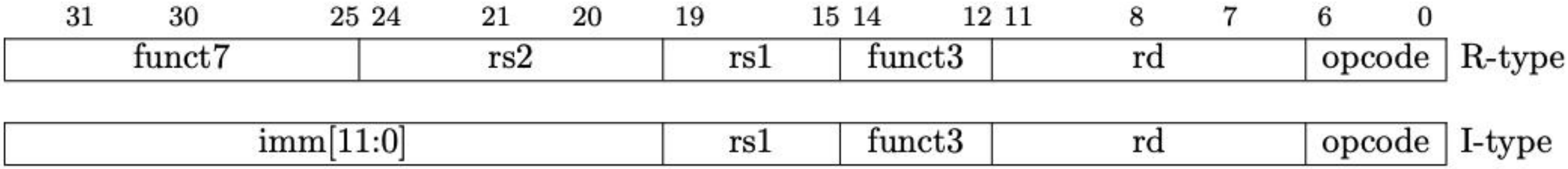
Datapath for R-type

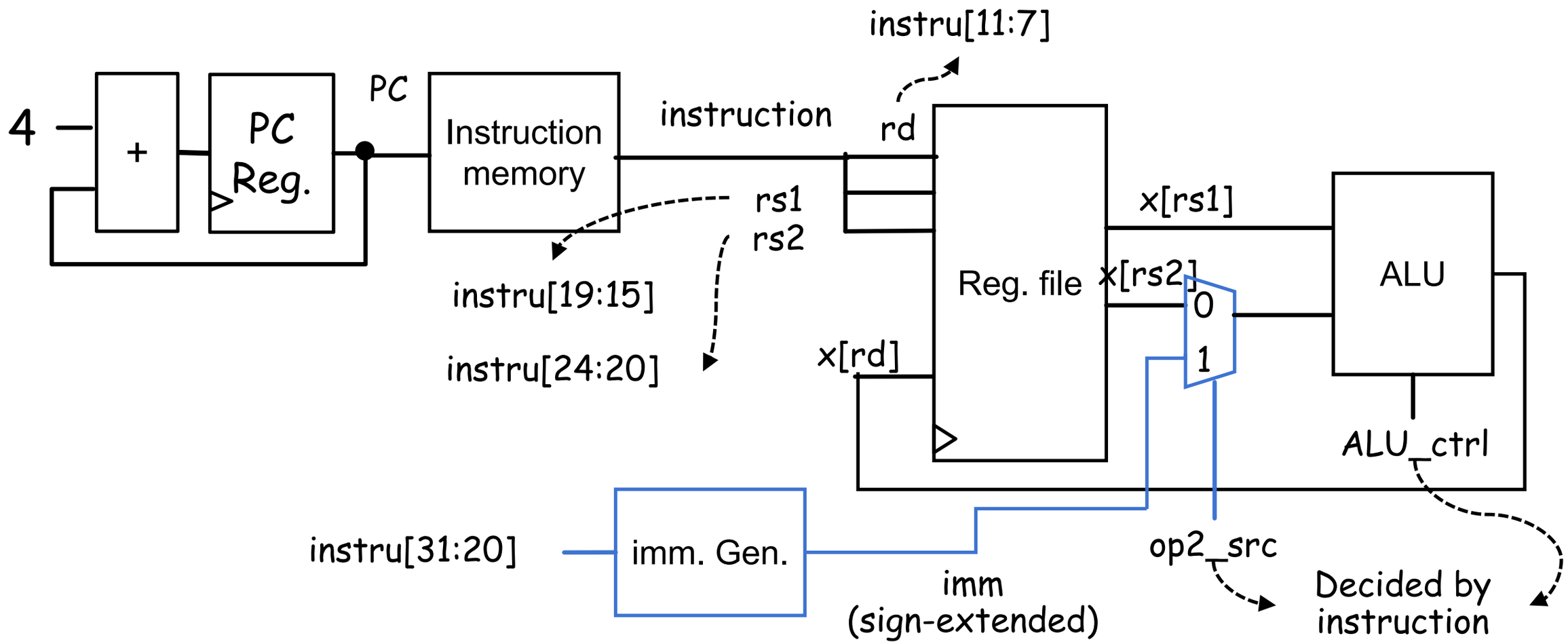
- We have all the building blocks to execute R-type instructions



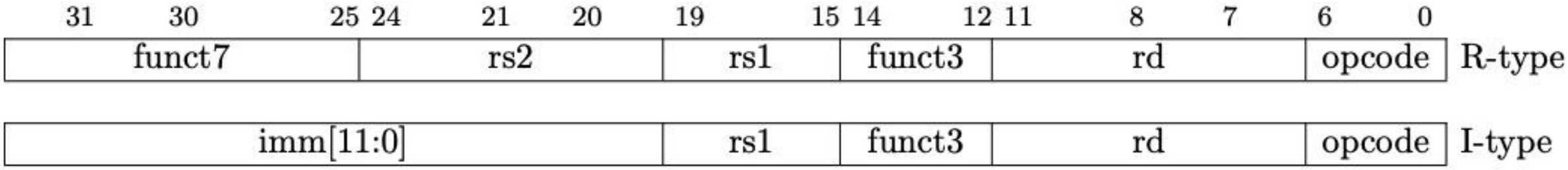


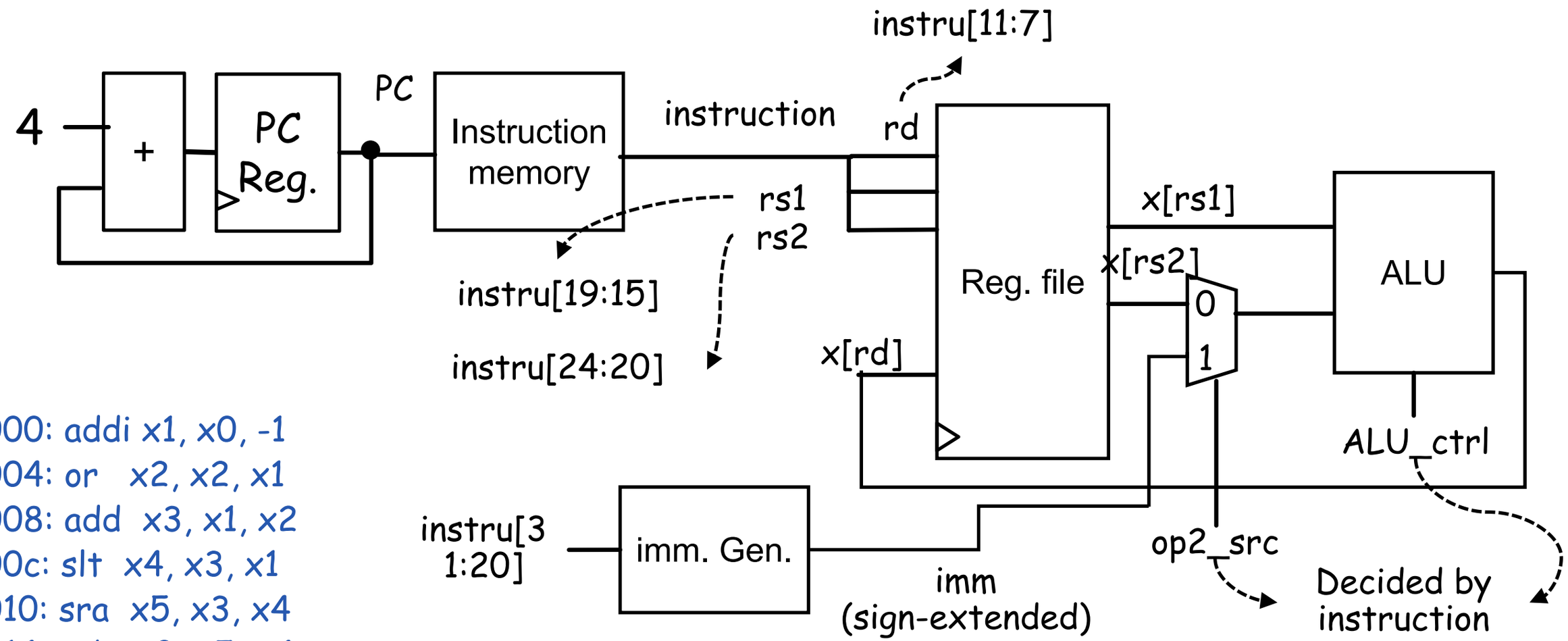
Datapath for I-type arithmetic and logic





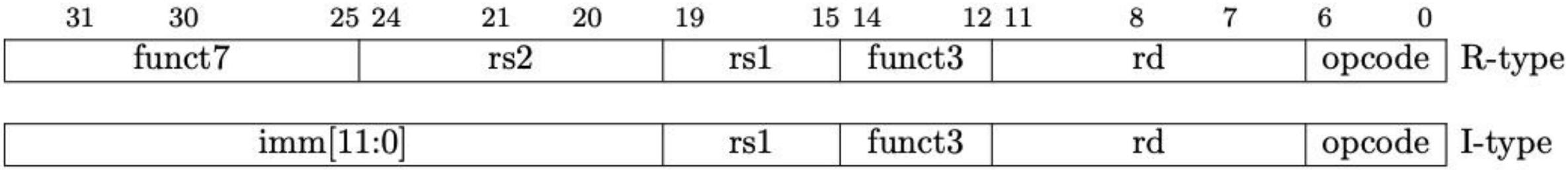
Example





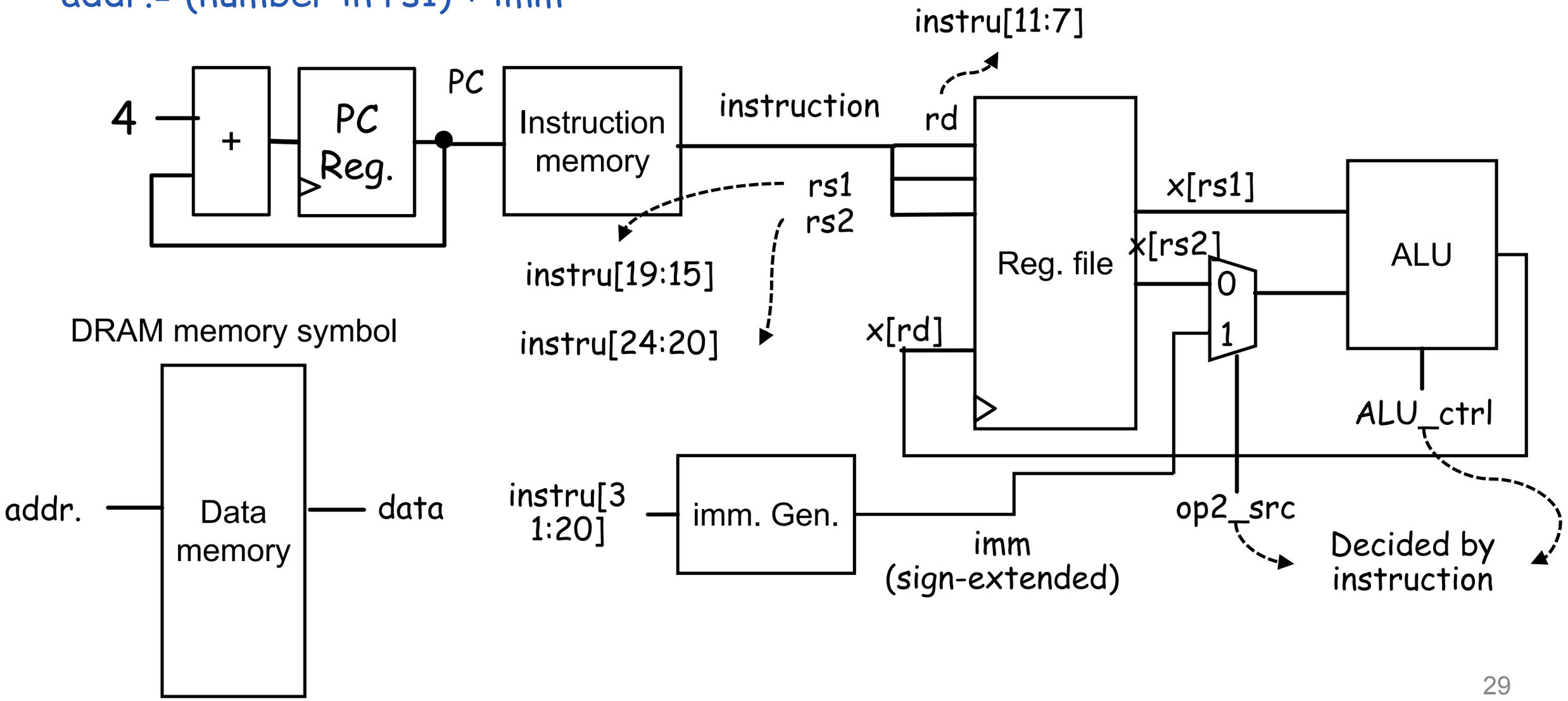
0x1000: addi x1, x0, -1
0x1004: or x2, x2, x1
0x1008: add x3, x1, x2
0x100c: slt x4, x3, x1
0x1010: sra x5, x3, x4
0x1014: sub x0, x5, x4

Datapath for more types ...

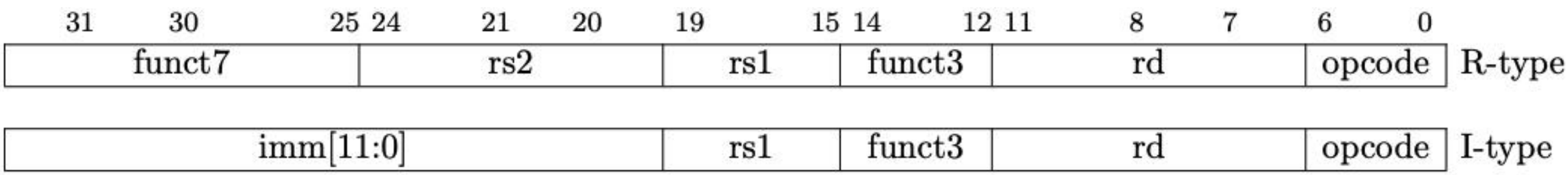


- lw rd, imm(rs1) : Load word at addr. to register rd

addr.= (number in rs1) + imm

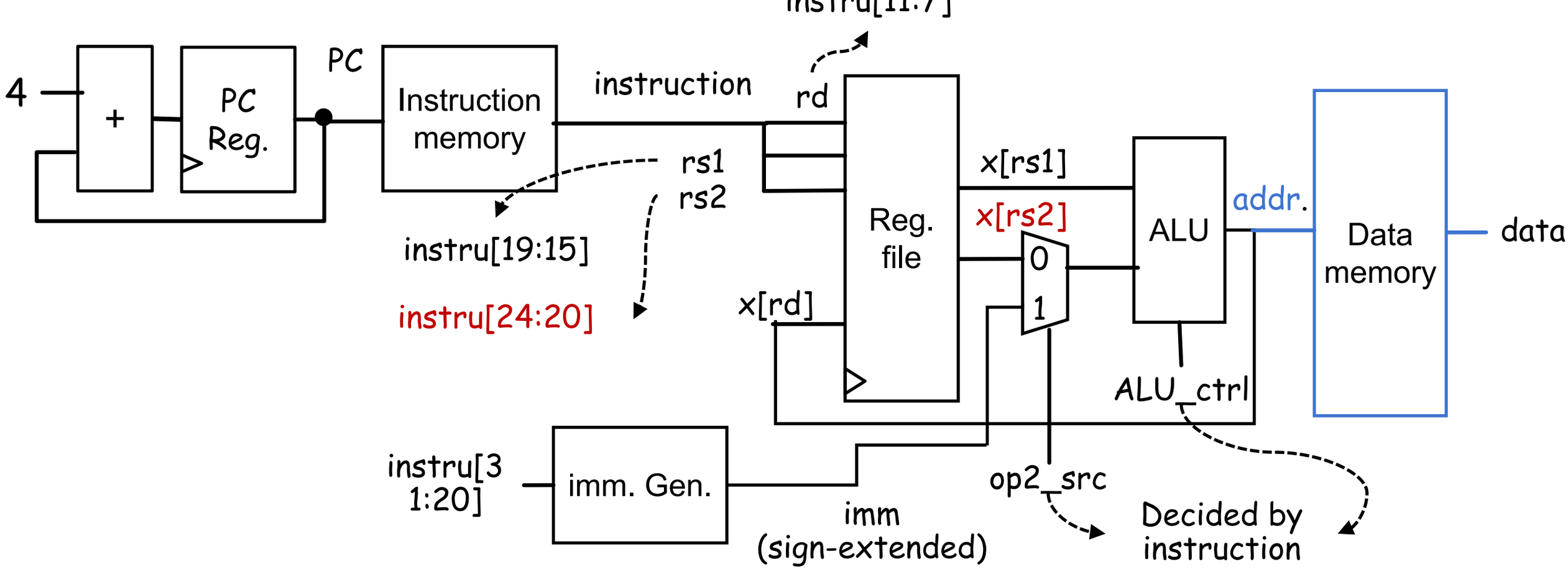


Datapath for I-type load

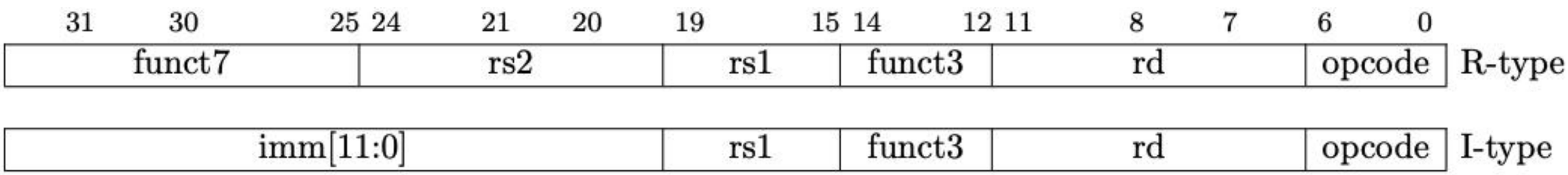


- lw rd, imm(rs1) : Load word at addr. to register rd

addr.= (number in rs1) + imm

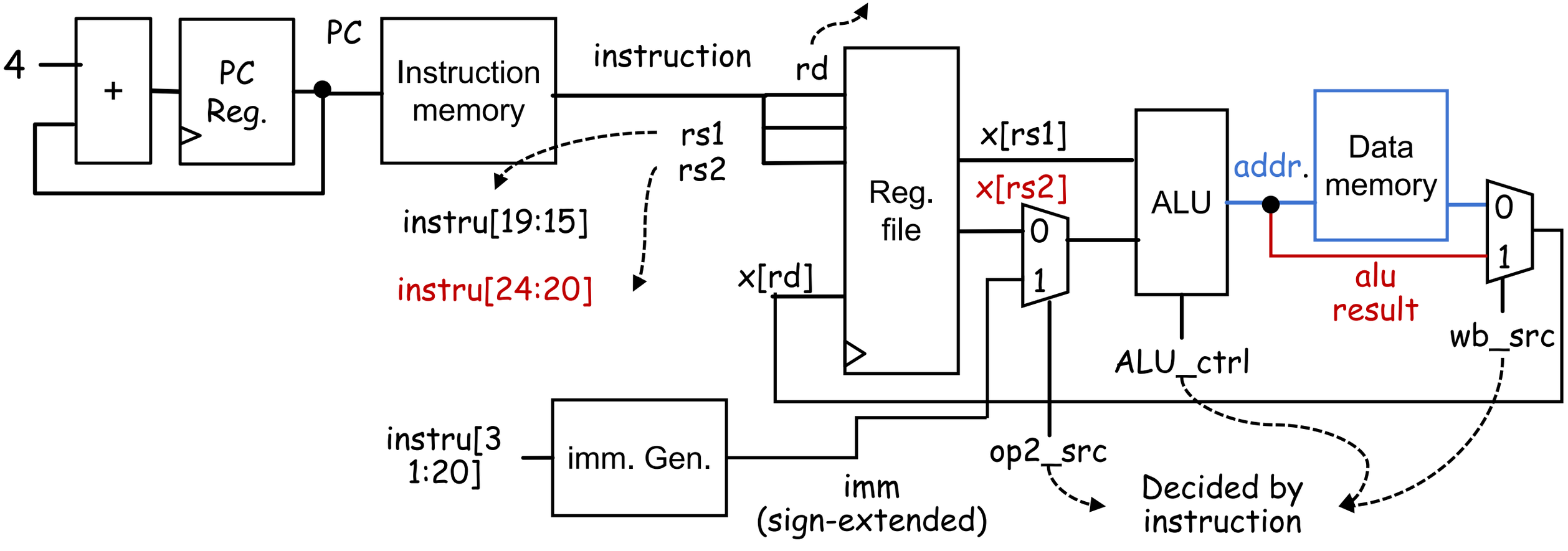


Datapath for I-type load

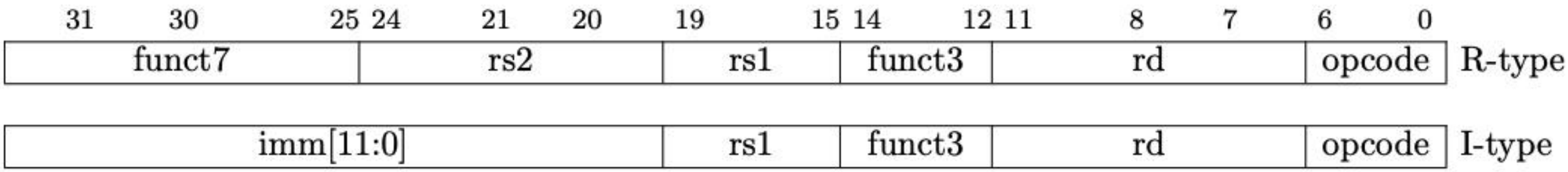


- lw rd, imm(rs1) : Load word at addr. to register rd

addr.= (number in rs1) + imm



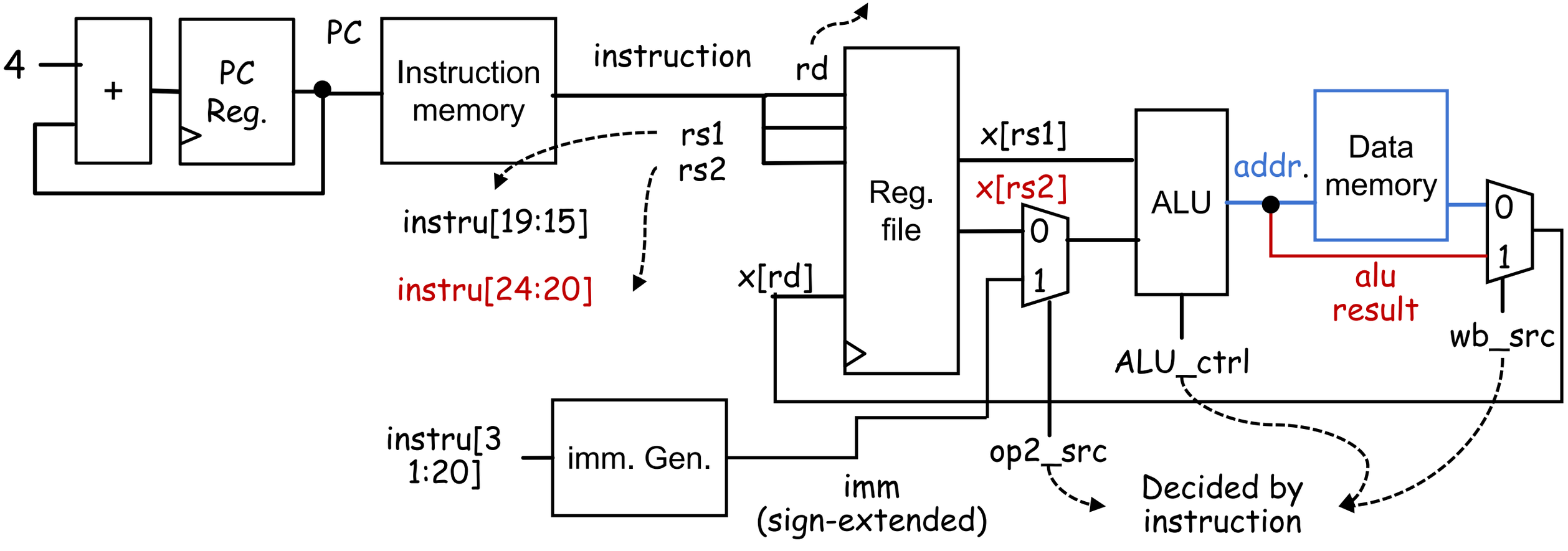
Datapath for I-type load



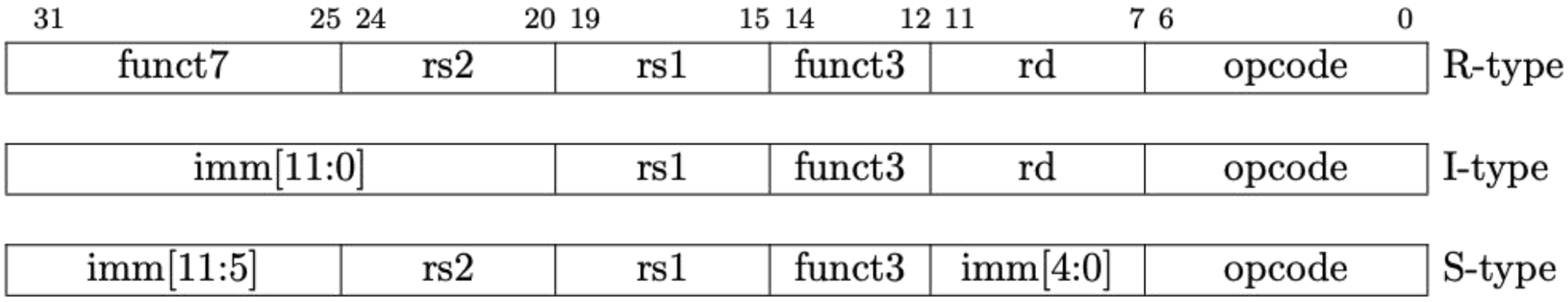
- lw rd, imm(rs1) : Load word at addr. to register rd

addr.= (number in rs1) + imm

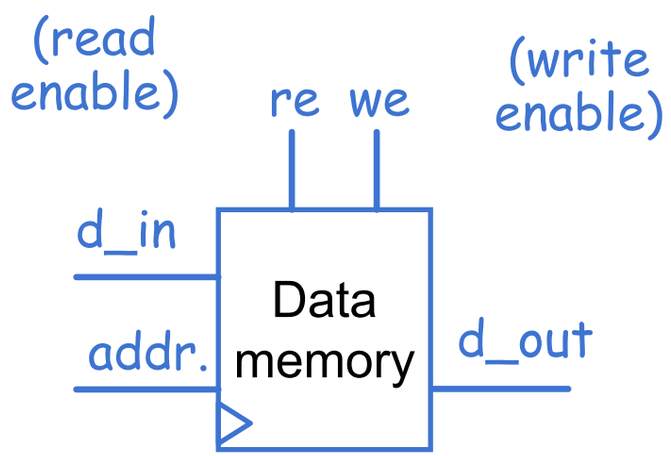
How would you deal with lb(u)/lh(u)?



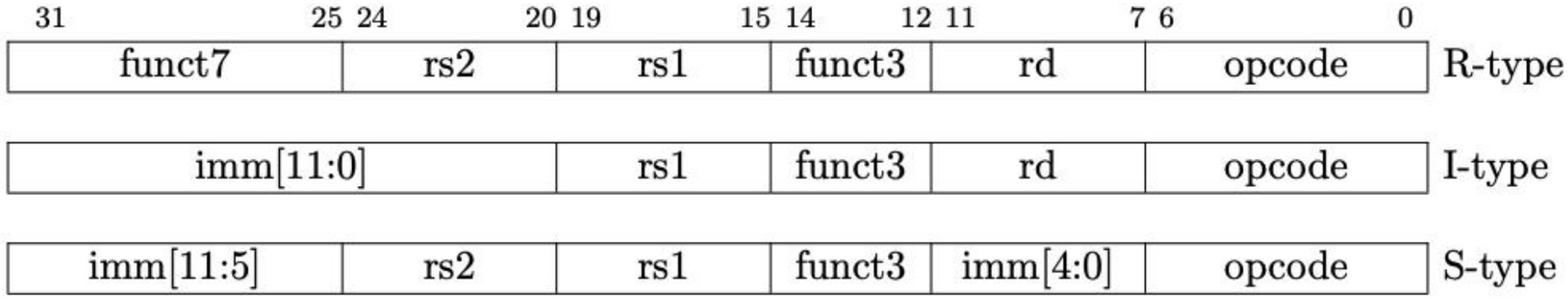
Datapath for S-type store



- Recall that in an FSM, only when there is a trigger (clk edge), the state can change.
- We assume that the change of data memory (memory-write) is also governed by clk edge.
- Assume behavior model of data memory:
  - When we=1 && re=0, at clk rising edge, data[addr.]=d_in; d_out stays at high-resistance ("z", output nothing)
  - When we=re=0, d_out stay at high-resistance (output nothing, state would not change); we=re=1 is forbidden
  - When we=0 && re=1, d_out=data[addr.]

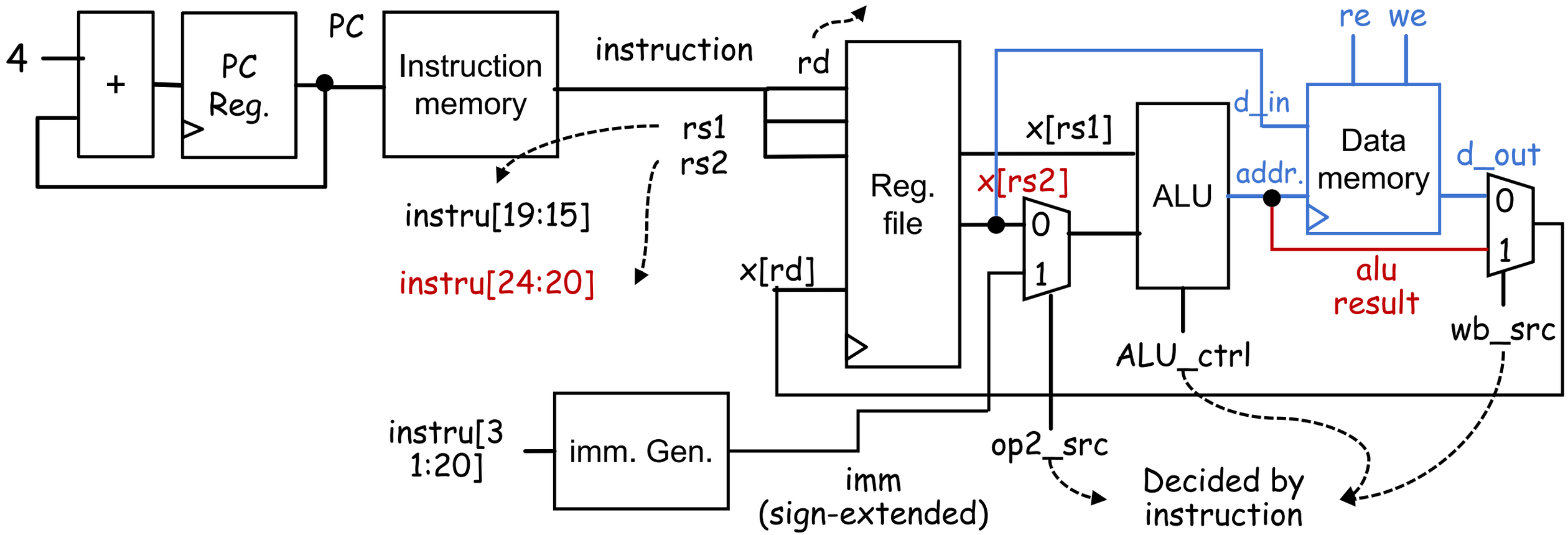


Datapath for S-type store

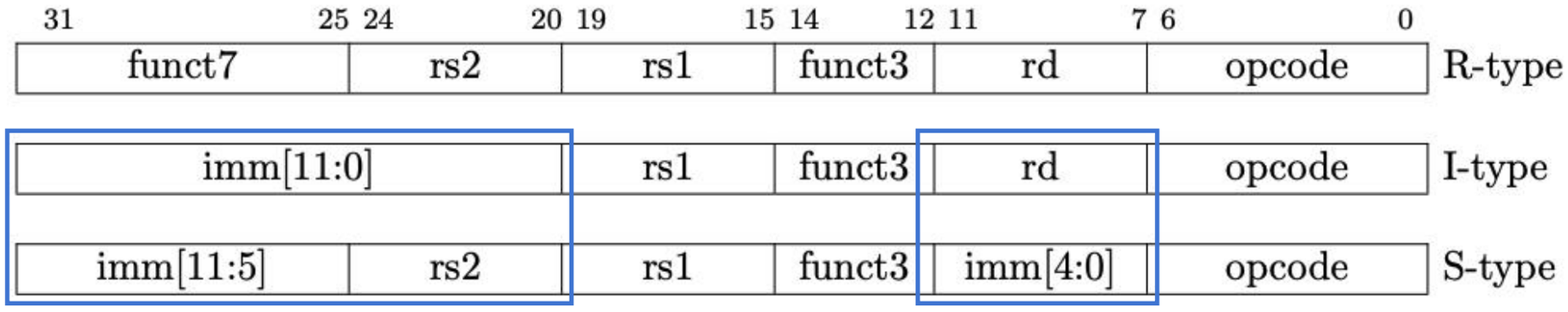


- sw rs2, imm(rs1): Store word in rs2 to memory addr.

addr.= (number in rs1) + imm

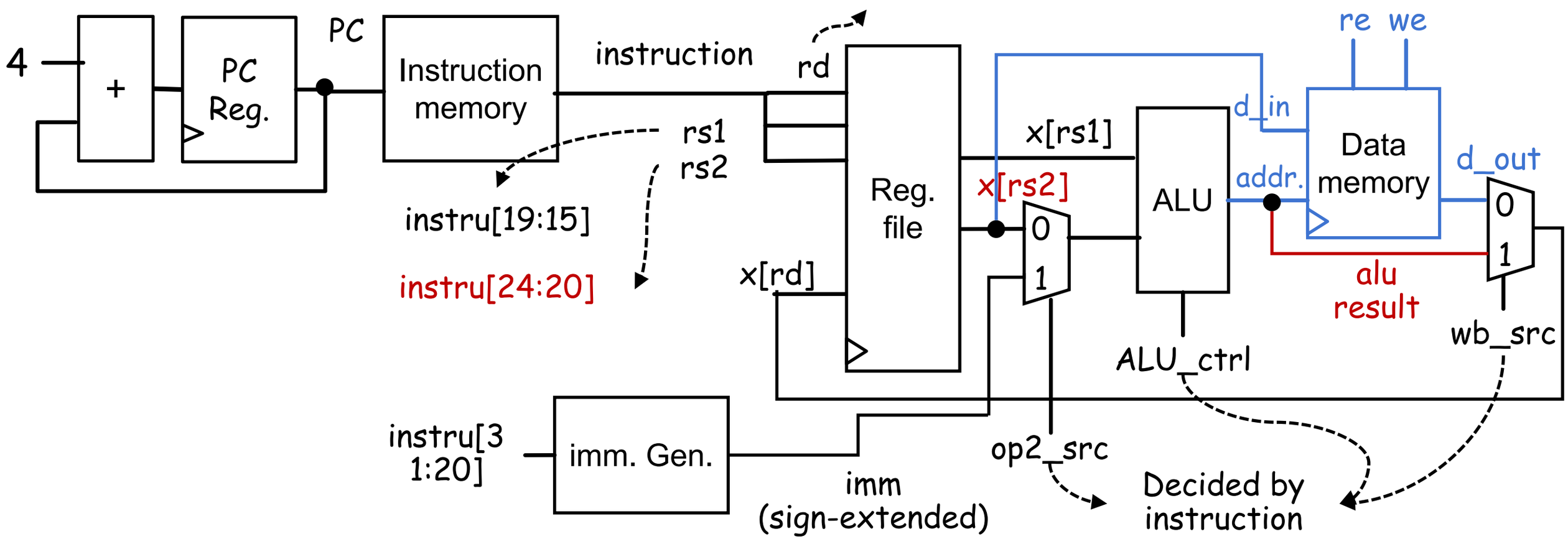


Datapath for S-type store

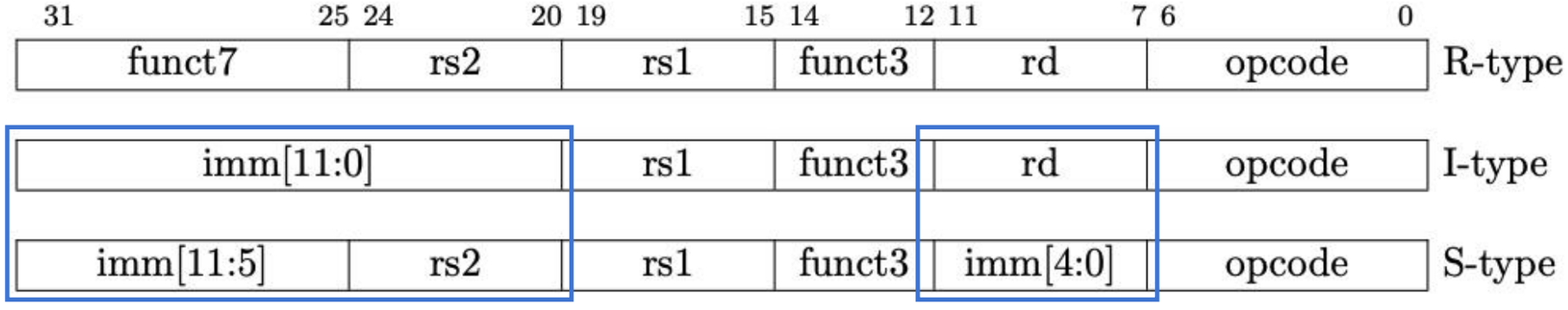


- sw rs2, imm(rs1): Store word at rs2 to memory addr.

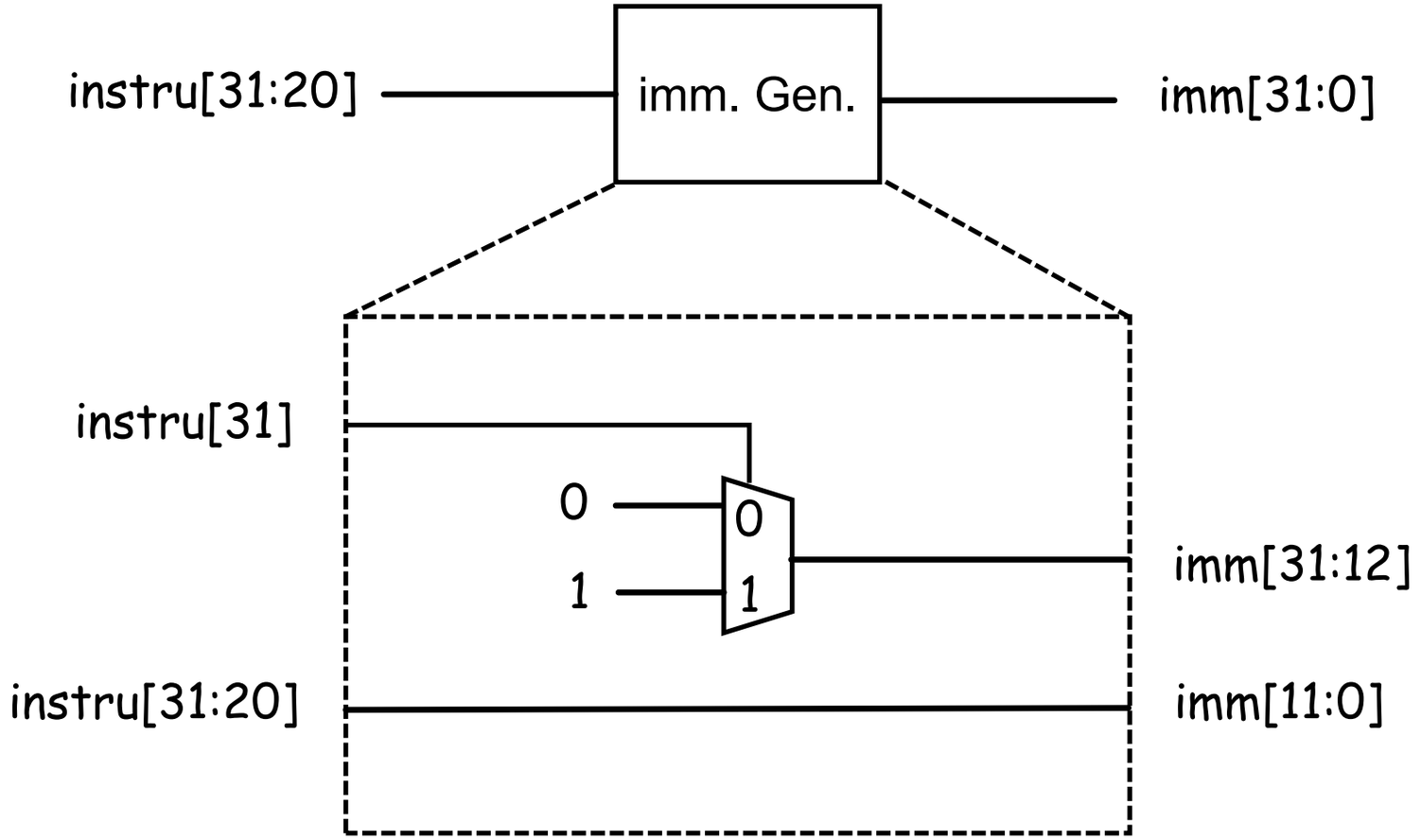
addr.= (number in rs1) + imm



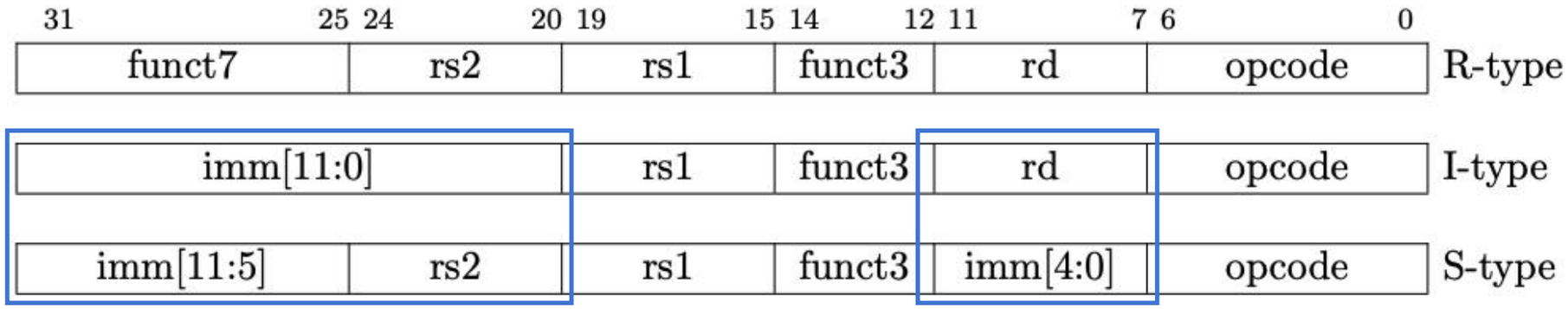
Imm. Gen.



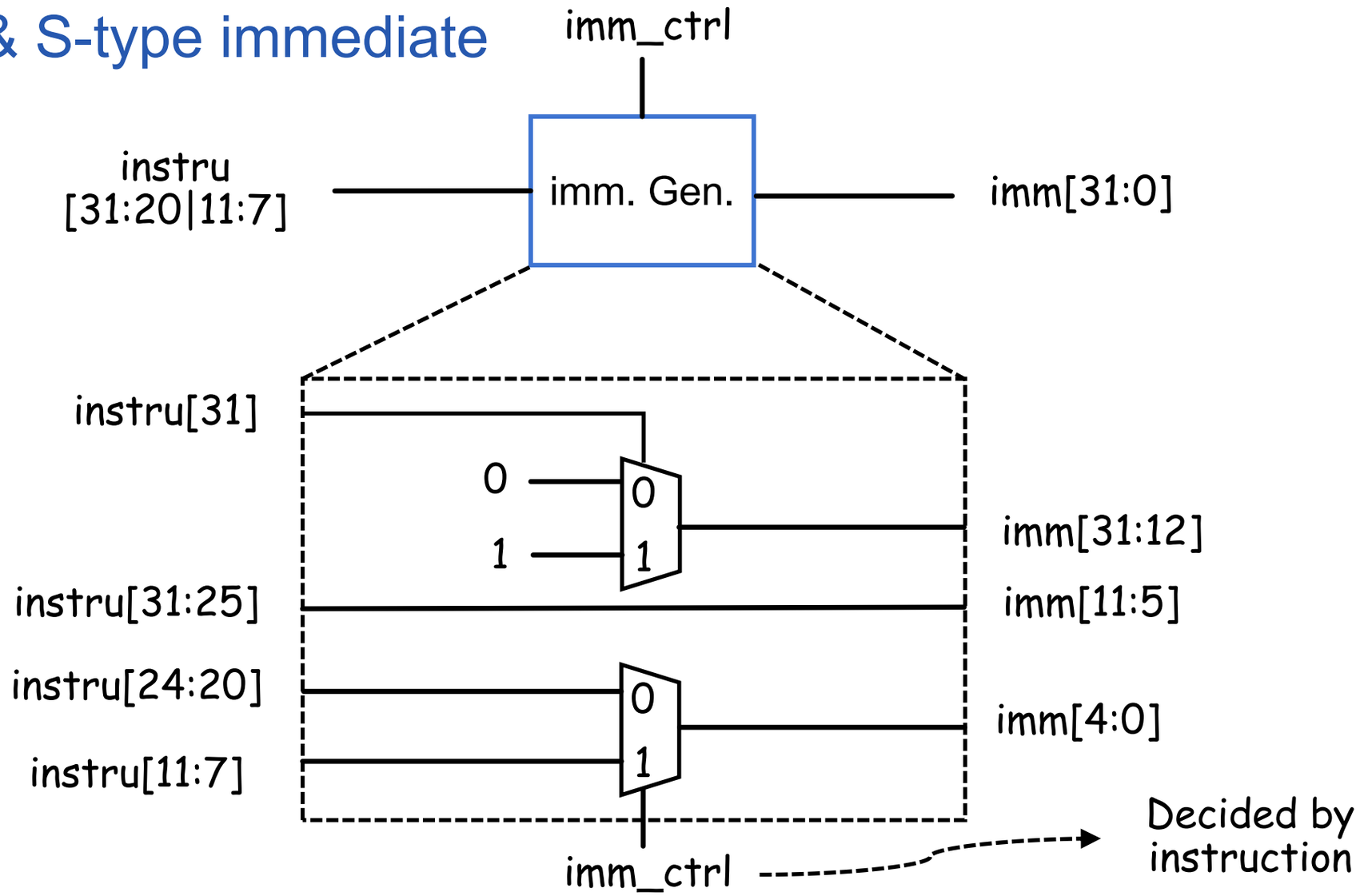
- I-type immediate



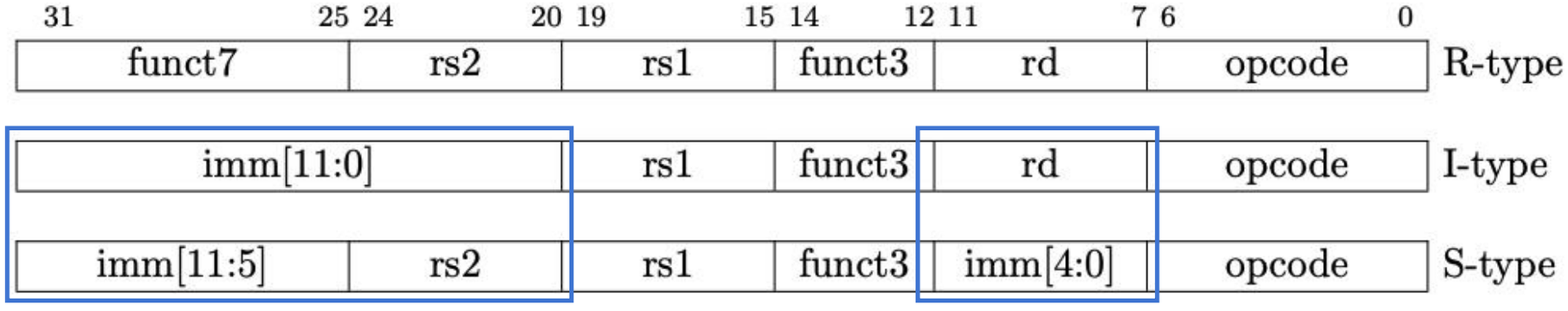
Imm. Gen.



- I-type & S-type immediate



Datapath for S-type store

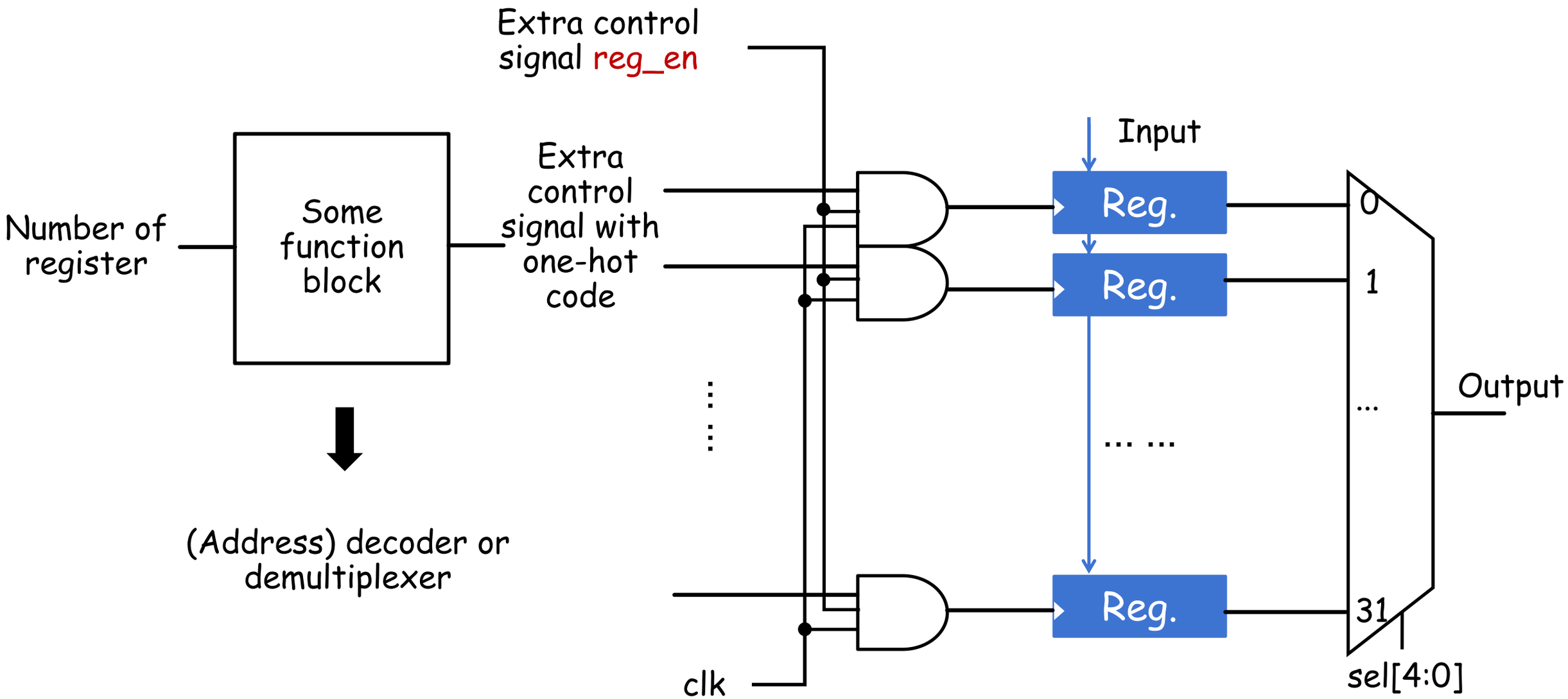


- sw rs2, imm(rs1): Store word at rs2 to memory addr.

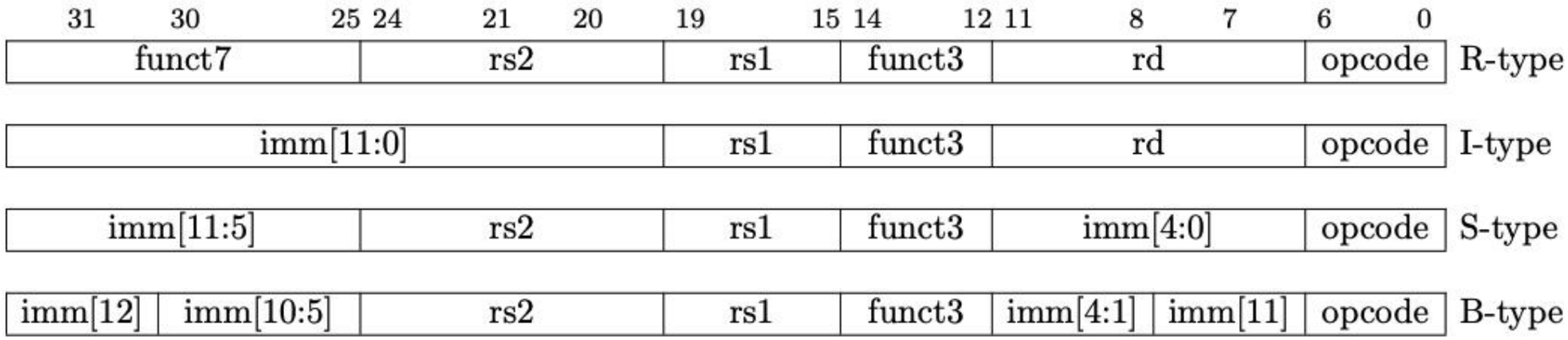
addr.= (number in rs1) + imm

Something wrong!

Regfile modification

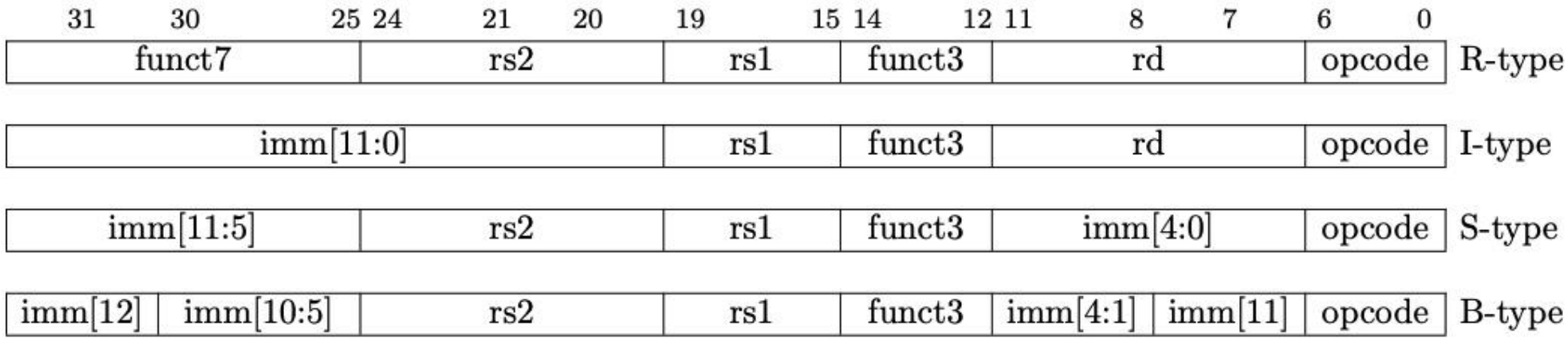


Datapath for B-type

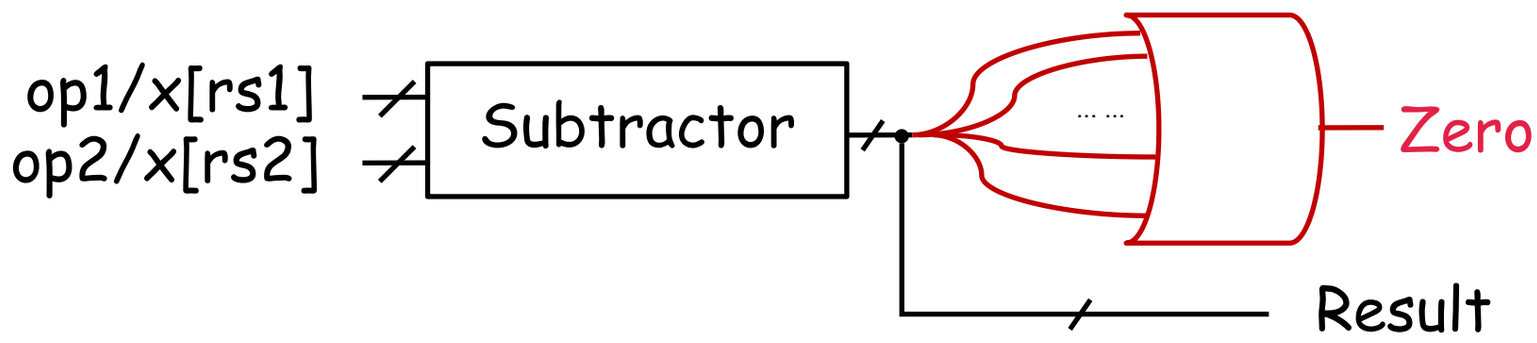


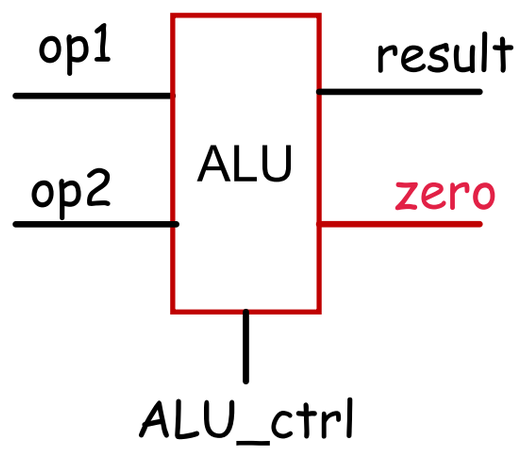
- beq rs1,rs2,L(imm/label)
- Go to label if x[rs1] == x[rs2]; otherwise, go to next statement

Datapath for B-type



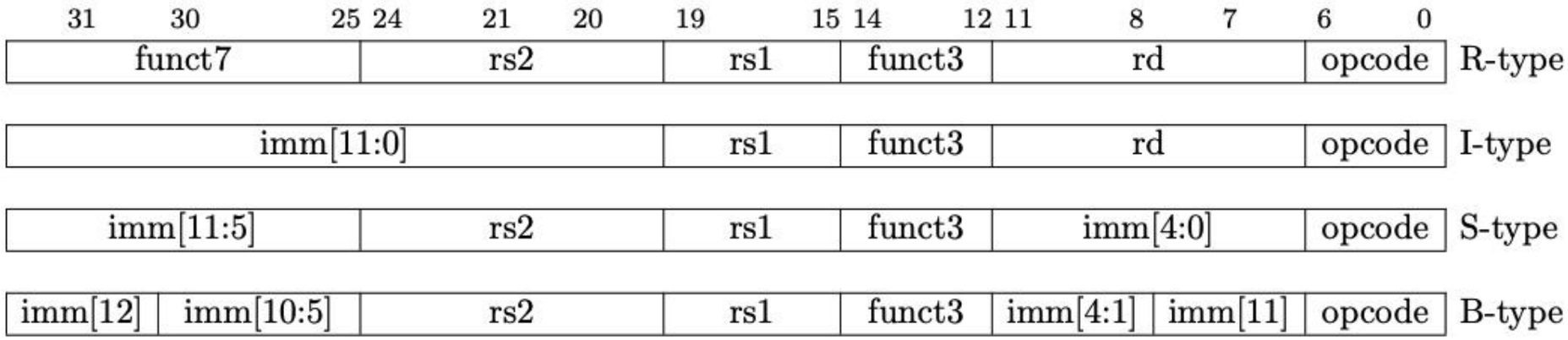
- beq rs1,rs2,L(imm/label)
- Go to label if x[rs1]==x[rs2]; otherwise, go to next statement
- Recall in ALU



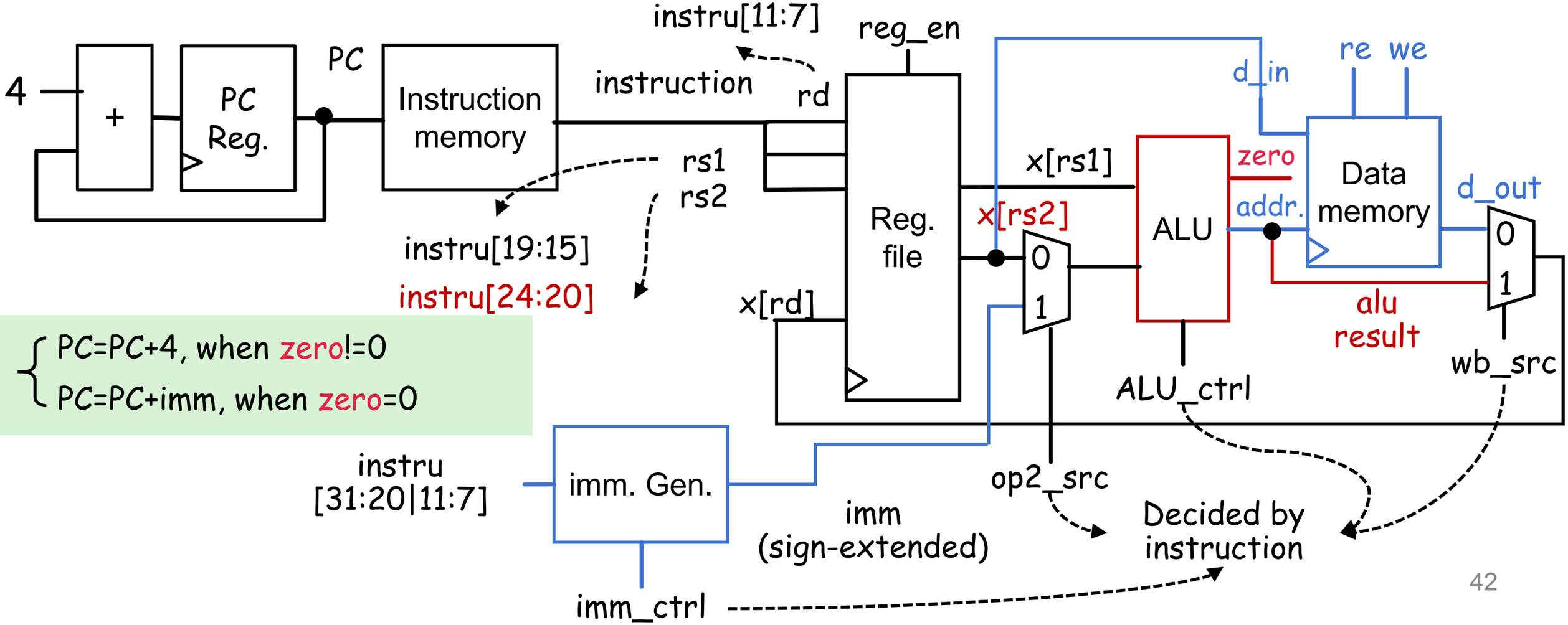


x[rs1]==x[rs2]↔ x[rs1]-x[rs2]==0

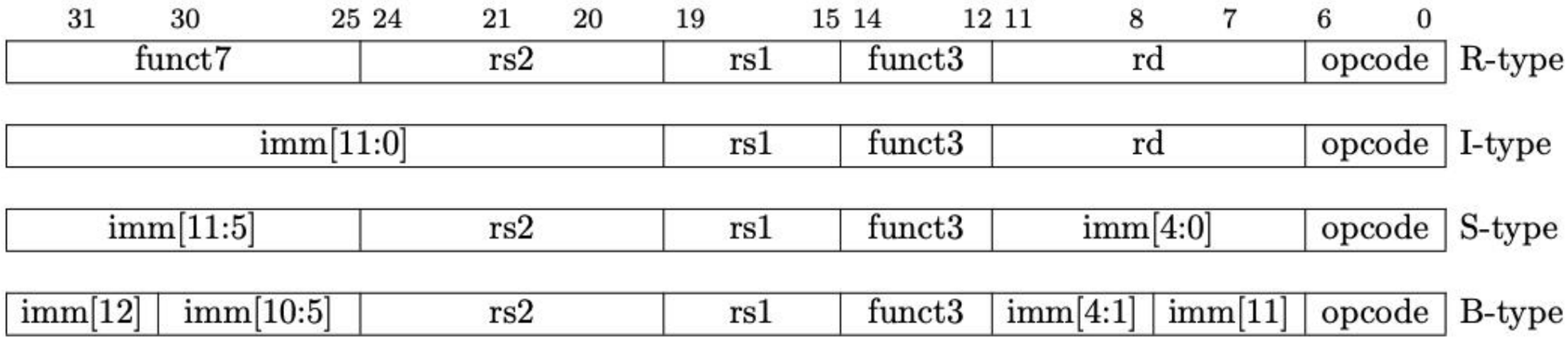
Datapath for B-type



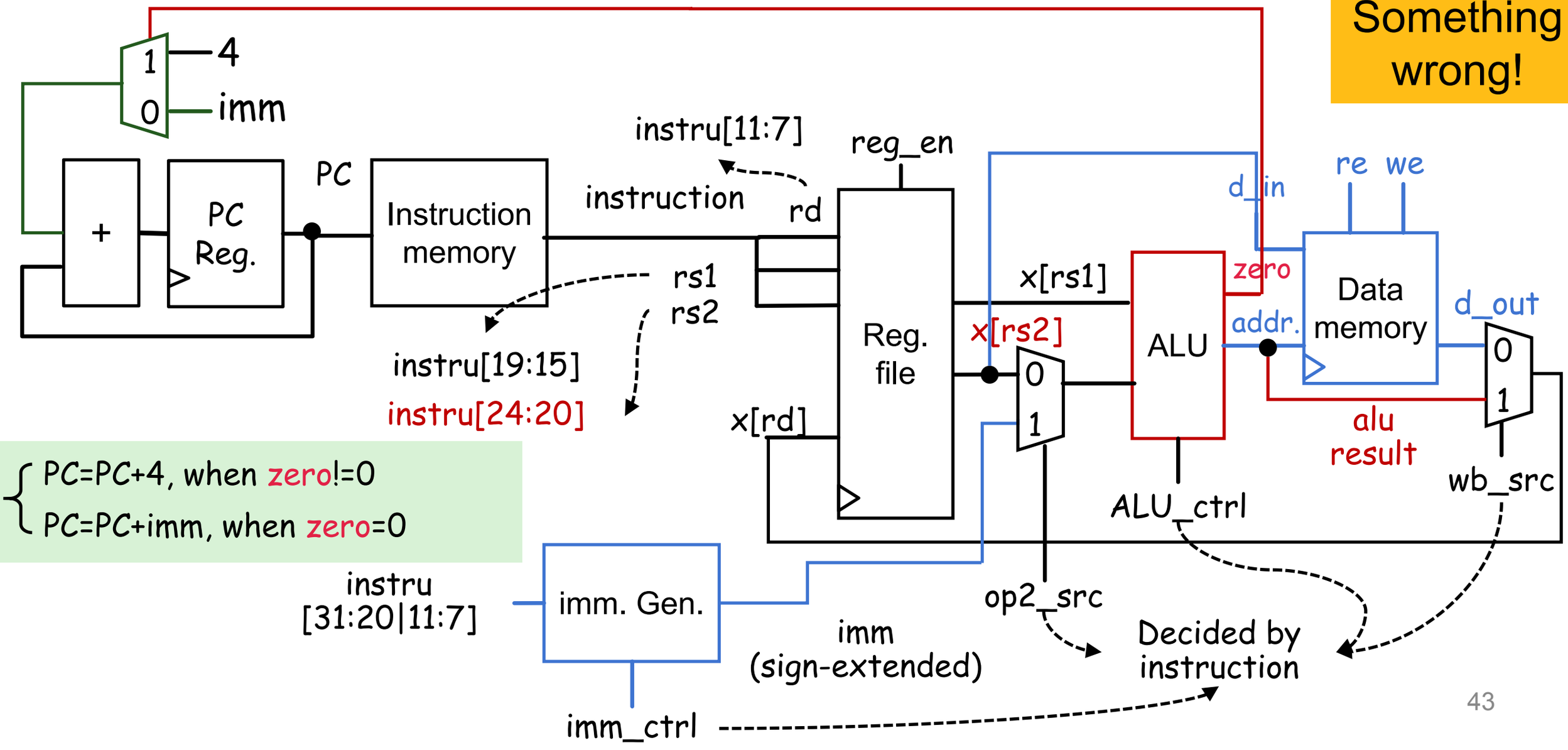
- beq rs1,rs2,L(imm/label)
- Go to label if x[rs1]==x[rs2]; otherwise, go to next instruction



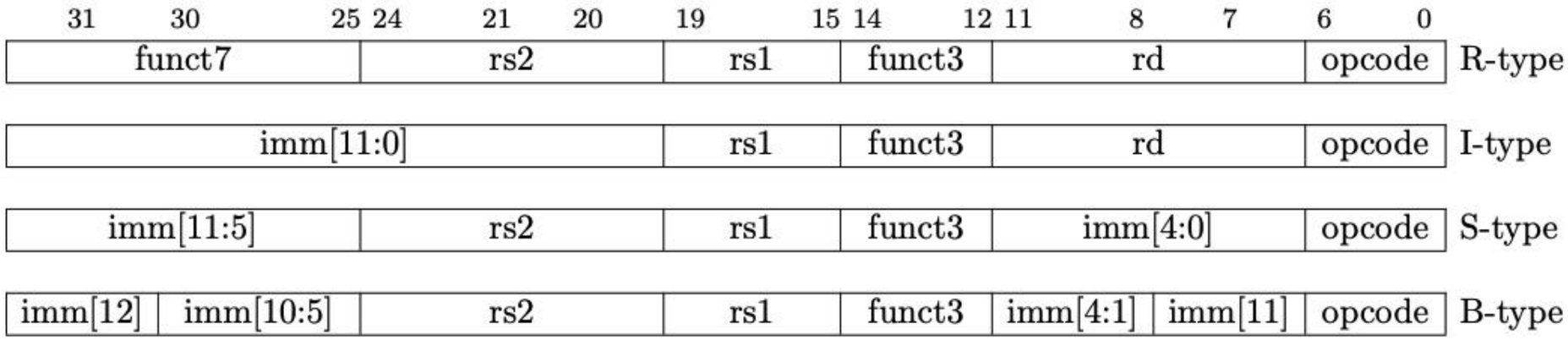
Datapath for B-type

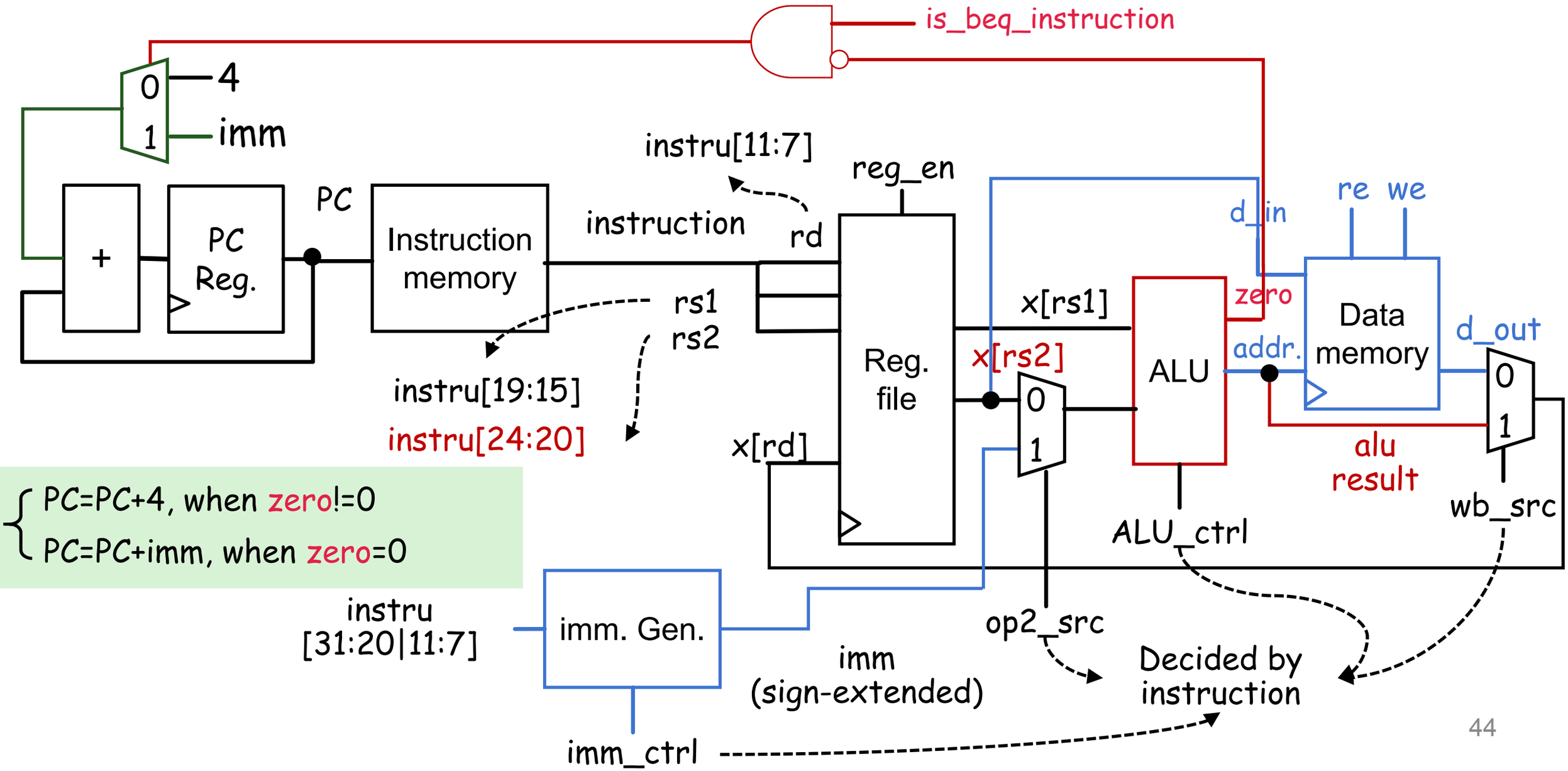


Something wrong!

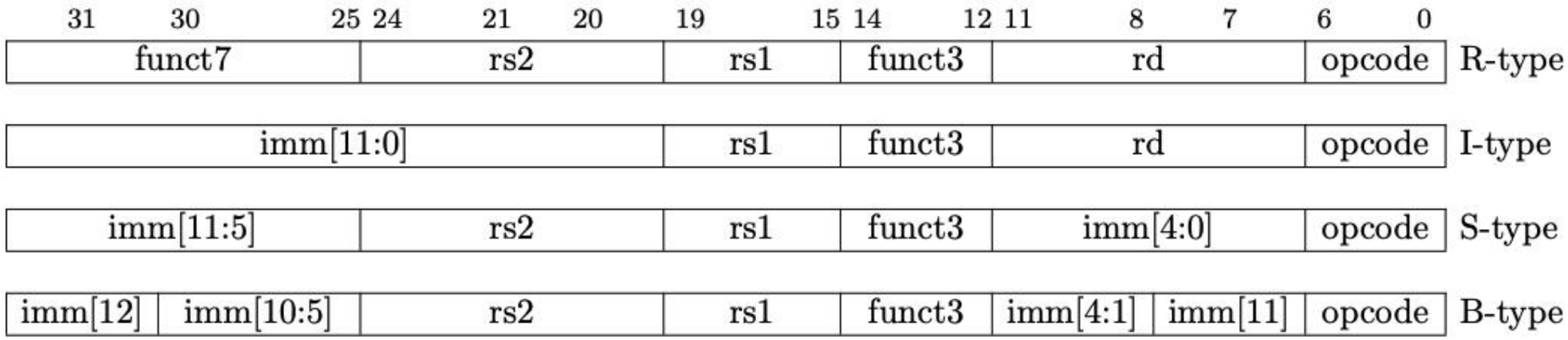


Datapath for B-type





Datapath for B-type



is_beq_instruction

- Recall beq

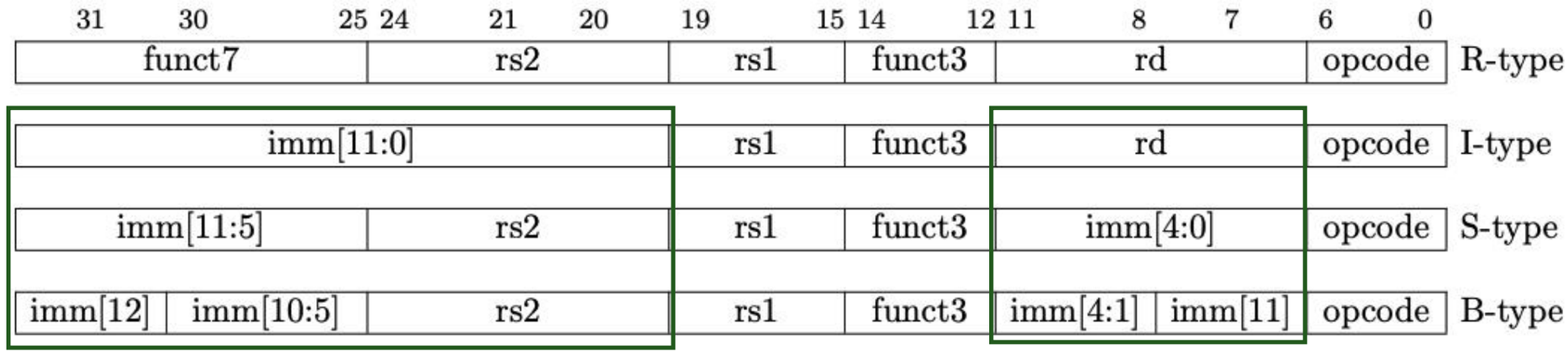
imm[12|10:5]	rs2	rs1	000	imm[4:1|11]	1100011	BEQ

Truth table

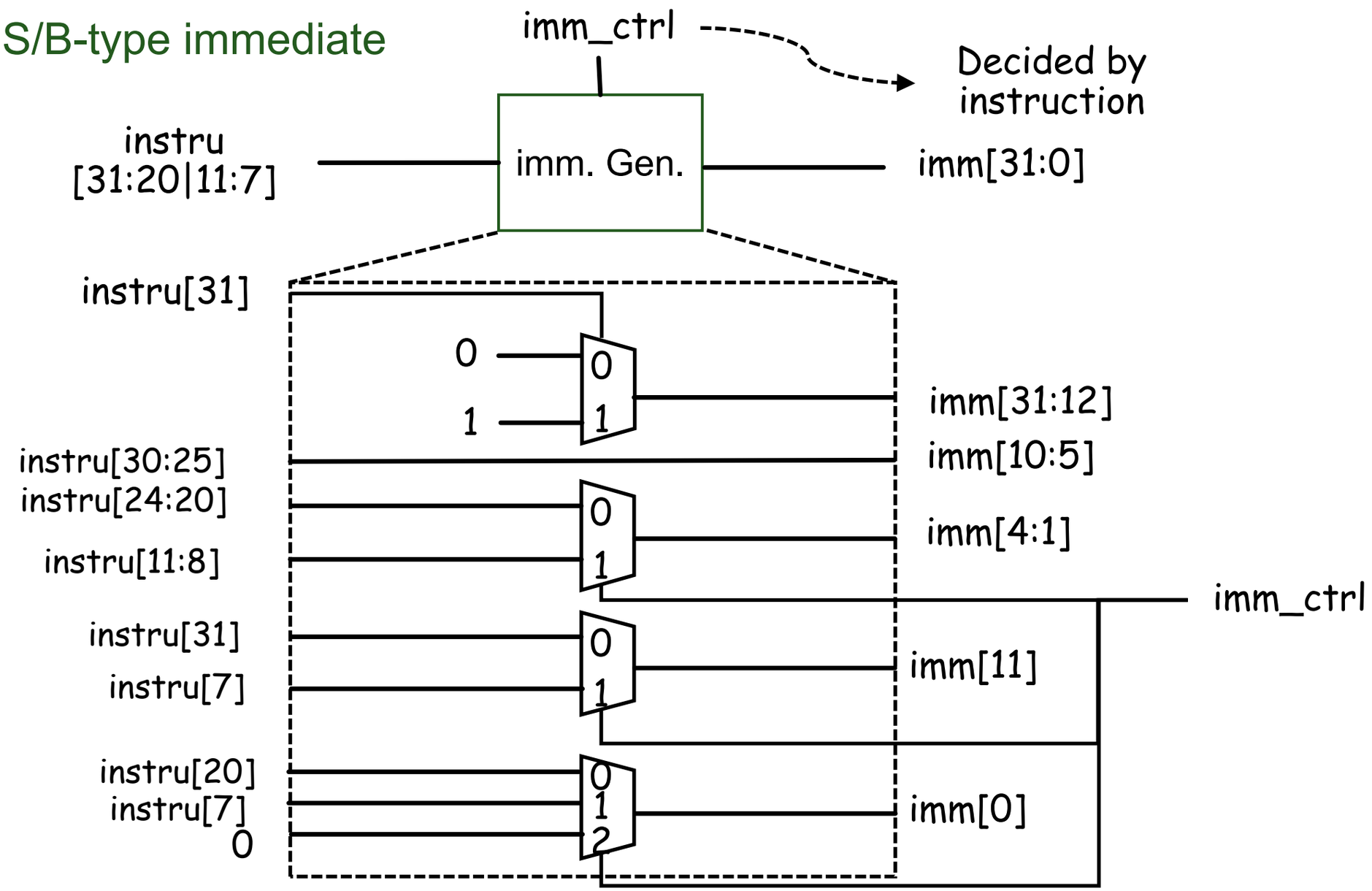
Instru[14:12|6:0]	is_beq_instruction
000 1100011	1
All the other cases	0

$$is\_beq\_instruction = i[14]\,\overline{i[13]}\,\overline{i[12]}\,i[6]\,i[5]\,\overline{i[4]}\,\overline{i[3]}\,\overline{i[2]}\,i[1]\,i[0]$$

Datapath for B-type

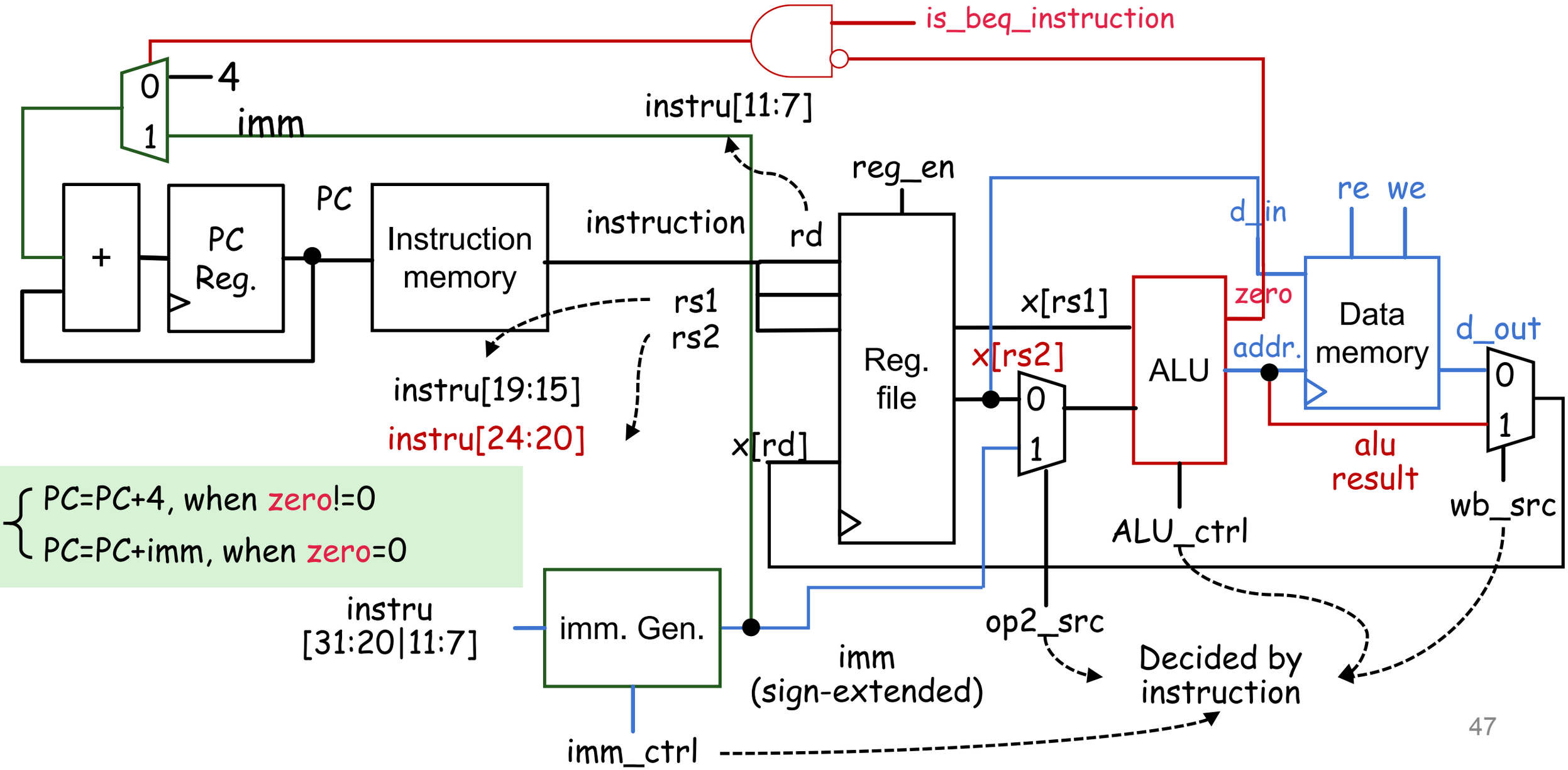


- I-type & S/B-type immediate

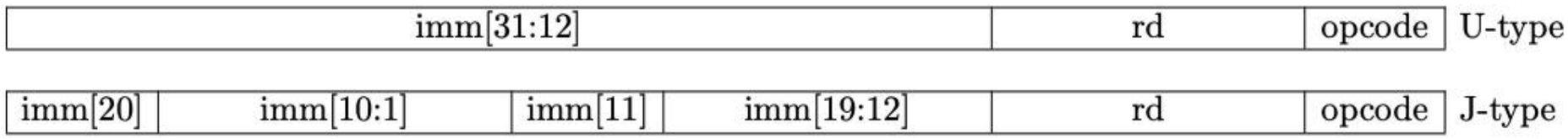


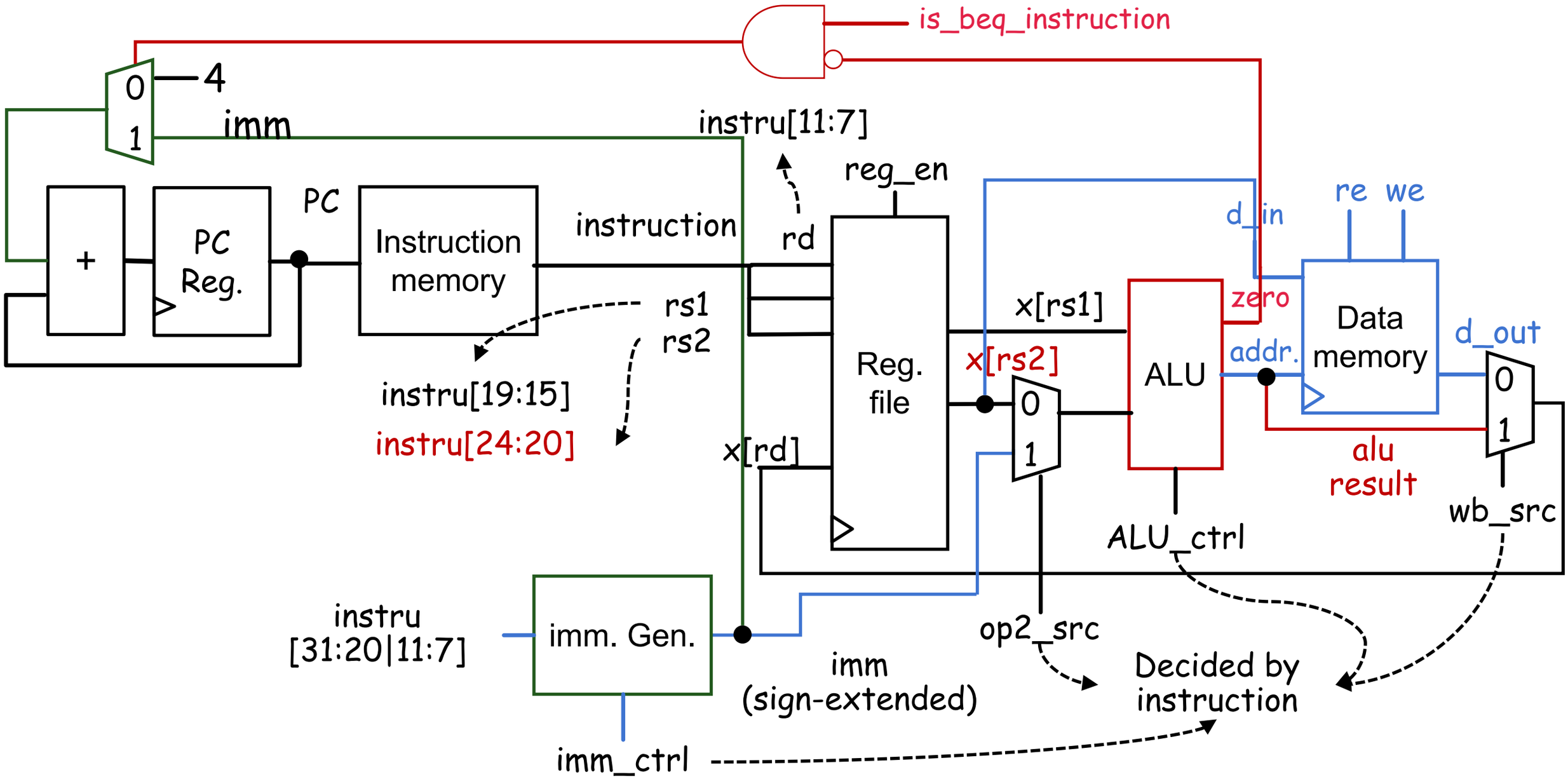
Datapath for B-type

31 30 25	24 21	20 19 15	14 12	11 8	7 6 0	
funct7	rs2	rs1	funct3	rd	opcode	R-type
imm[11:0]		rs1	funct3	rd	opcode	I-type
imm[11:5]	rs2	rs1	funct3	imm[4:0]	opcode	S-type
imm[12] imm[10:5]	rs2	rs1	funct3	imm[4:1] imm[11]	opcode	B-type



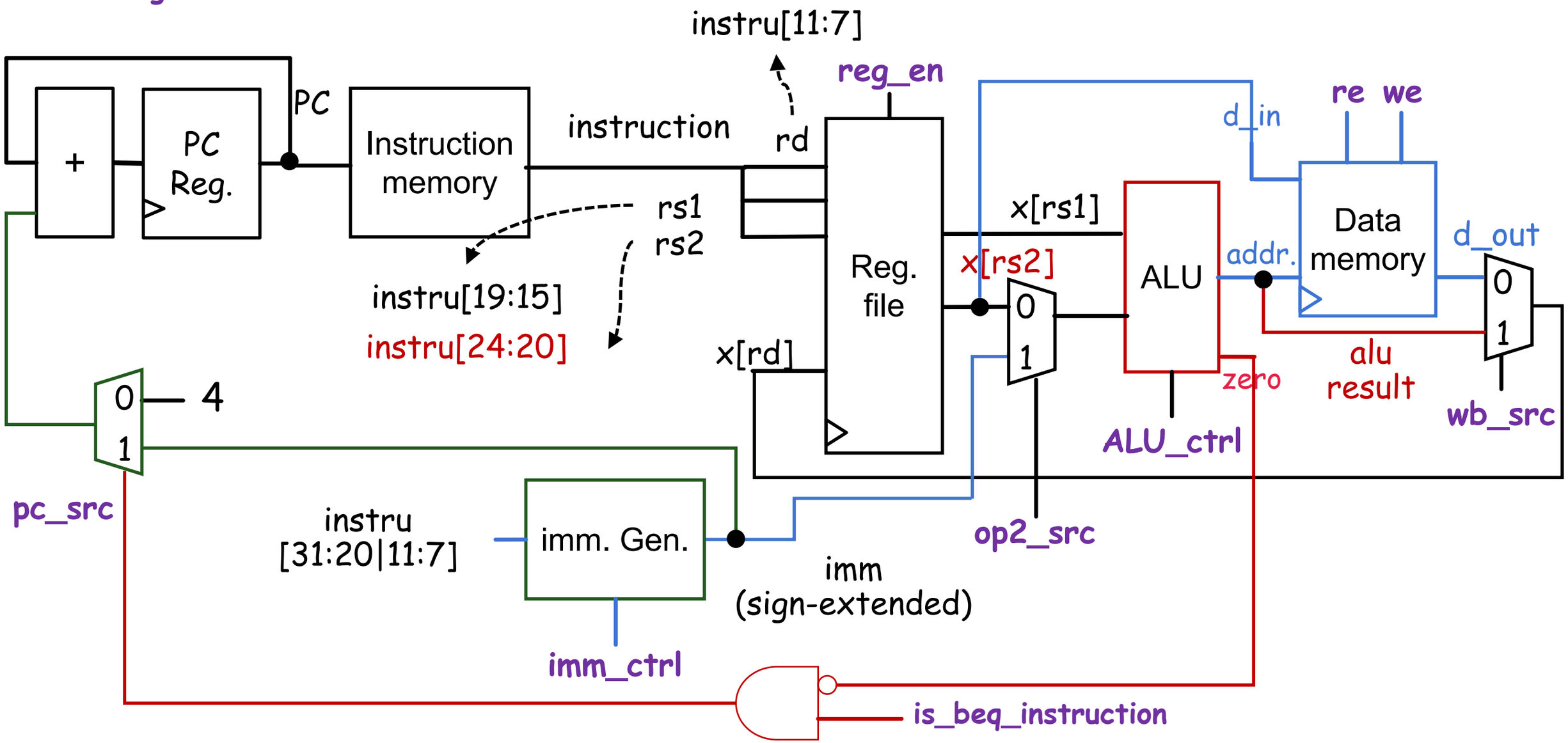
Datapath for the other types



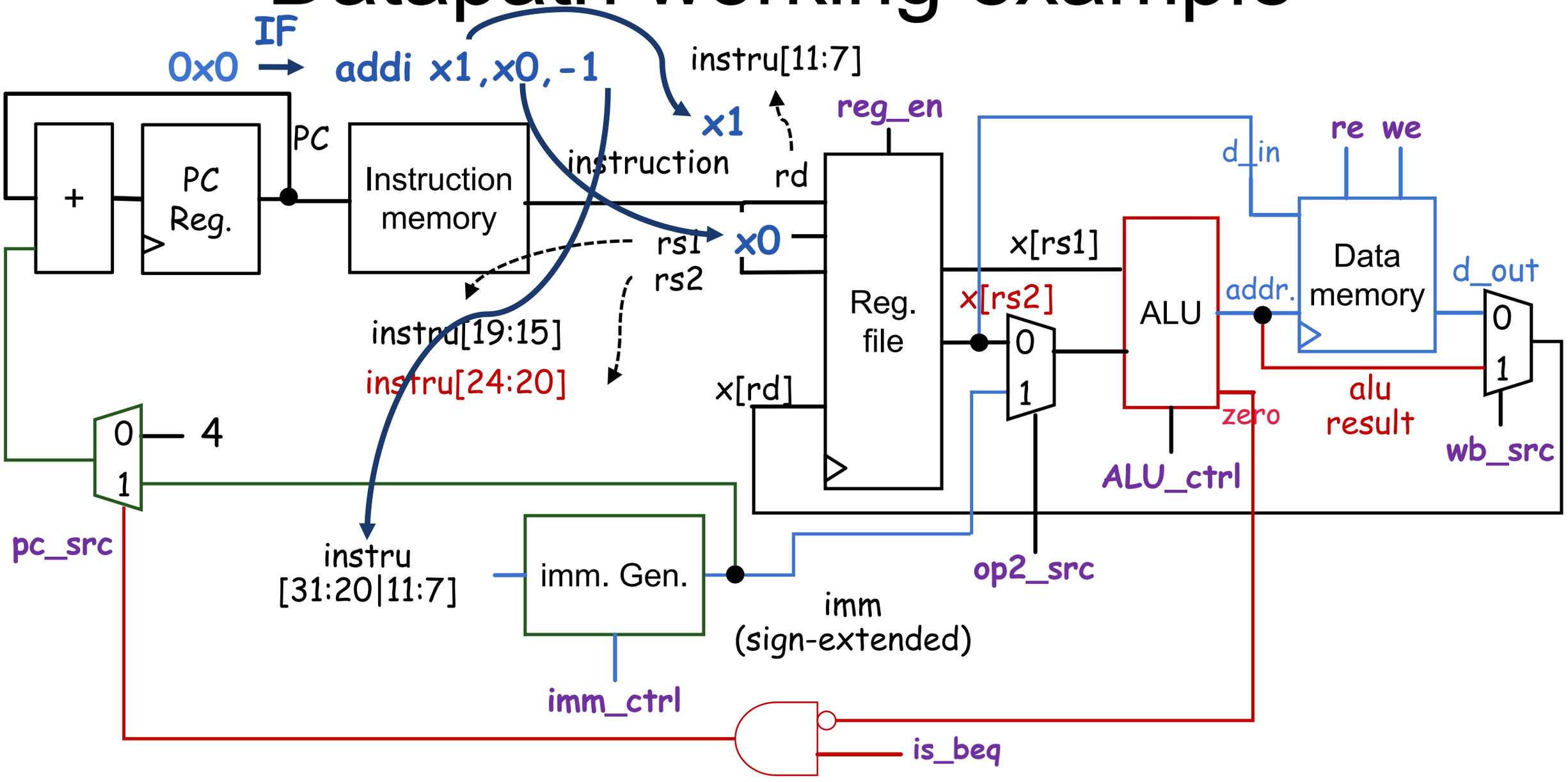


Control signals

- This is a datapath that supports R-type & I-type arithmetic and logic operations, *lw*, *sw* and *beq*



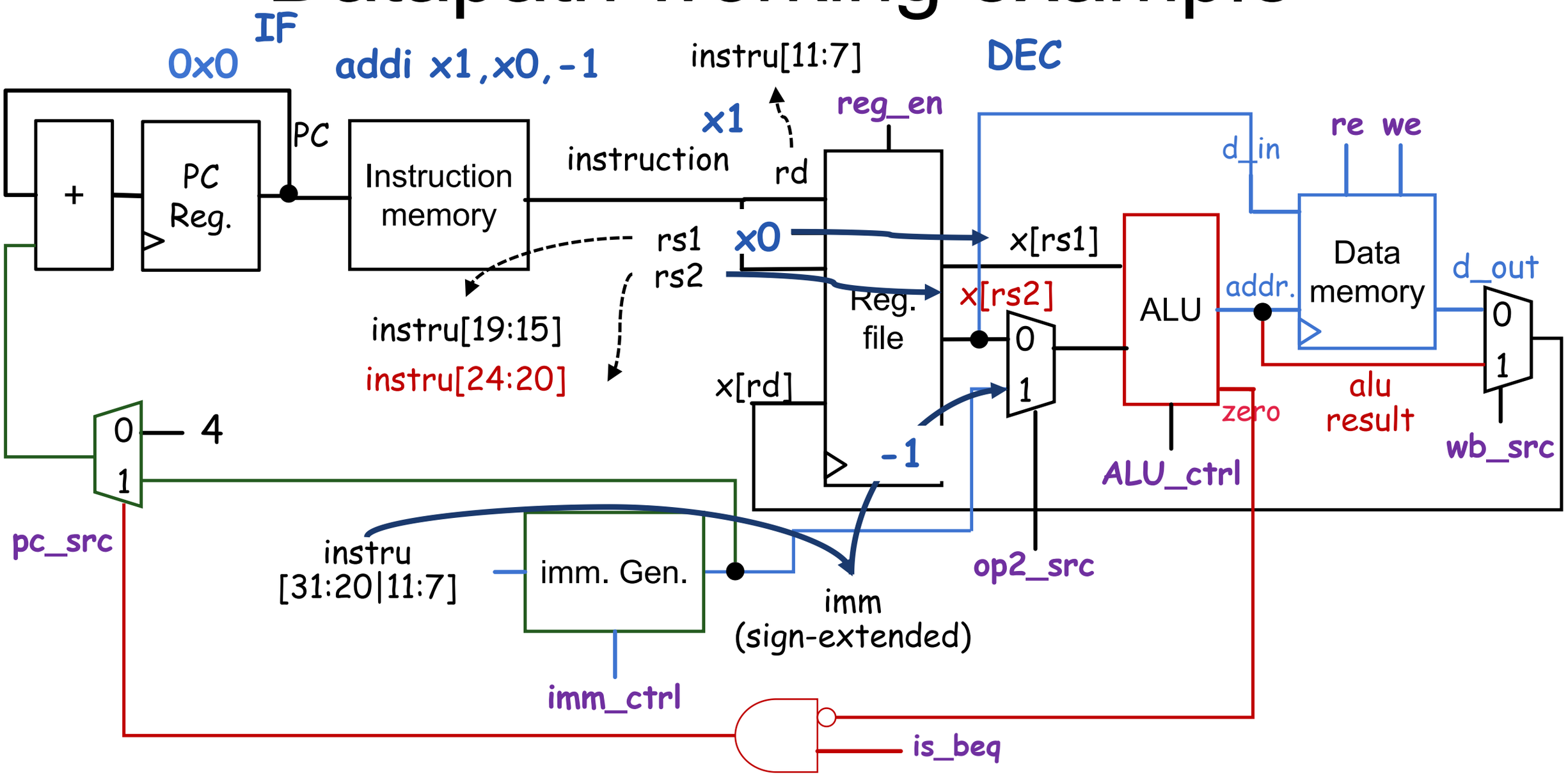
Datapath working example



	reg_en	re	we	alu_ctrl	imm_ctrl	wb_src	op2_src	is_beq
addi	1	0	0	add	I-type	1	1	0

0x0: addi x1, x0, -1
0x4: ori x2, x0, 128
0x8: add x3, x1, x2
0xc: sw x3, 0(x3)
0x10:lw x5, 0(x3)
0x14:beq x3, x5, -12

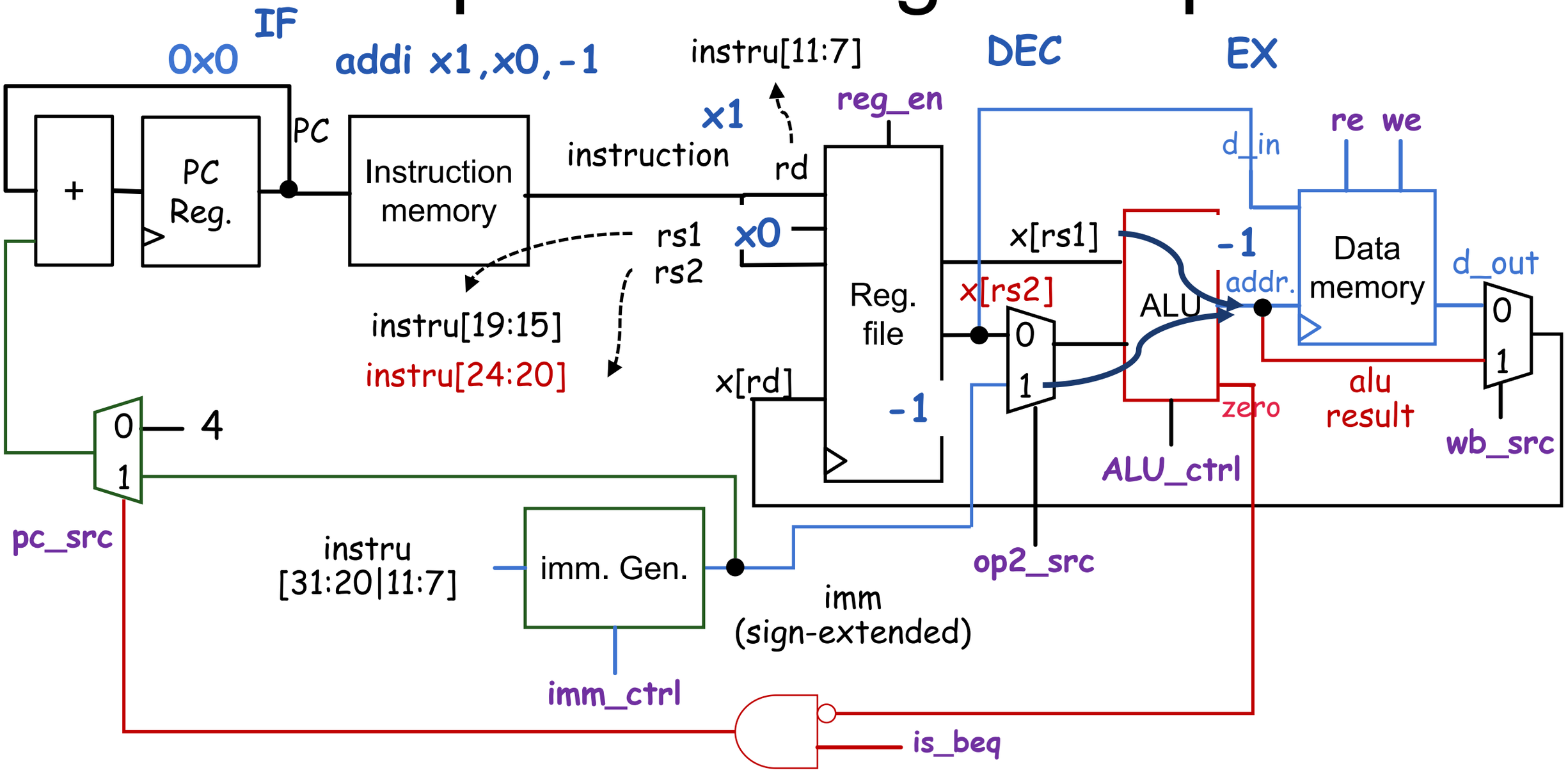
Datapath working example



	reg_en	re	we	alu_ctrl	imm_ctrl	wb_src	op2_src	is_beq
addi	1	0	0	add	I-type	1	1	0

0x0: addi x1, x0, -1
0x4: ori x2, x0, 128
0x8: add x3, x1, x2
0xc: sw x3, 0(x3)
0x10:lw x5, 0(x3)
0x14:beq x3, x5, -12

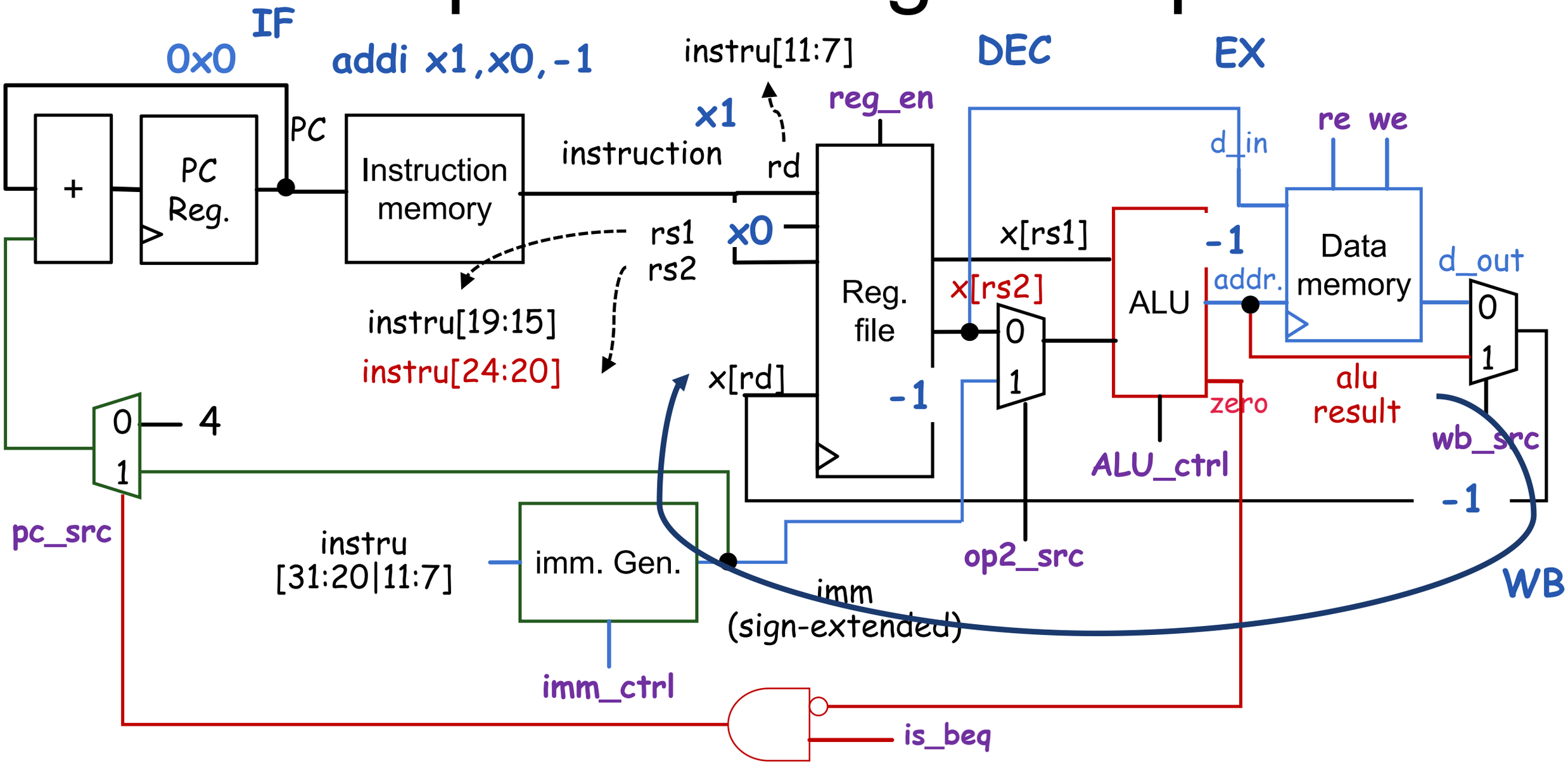
Datapath working example



	reg_en	re	we	alu_ctrl	imm_ctrl	wb_src	op2_src	is_beq
addi	1	0	0	add	I-type	1	1	0

0x0: addi x1, x0, -1
0x4: ori x2, x0, 128
0x8: add x3, x1, x2
0xc: sw x3, 0(x3)
0x10:lw x5, 0(x3)
0x14:beq x3, x5, -12

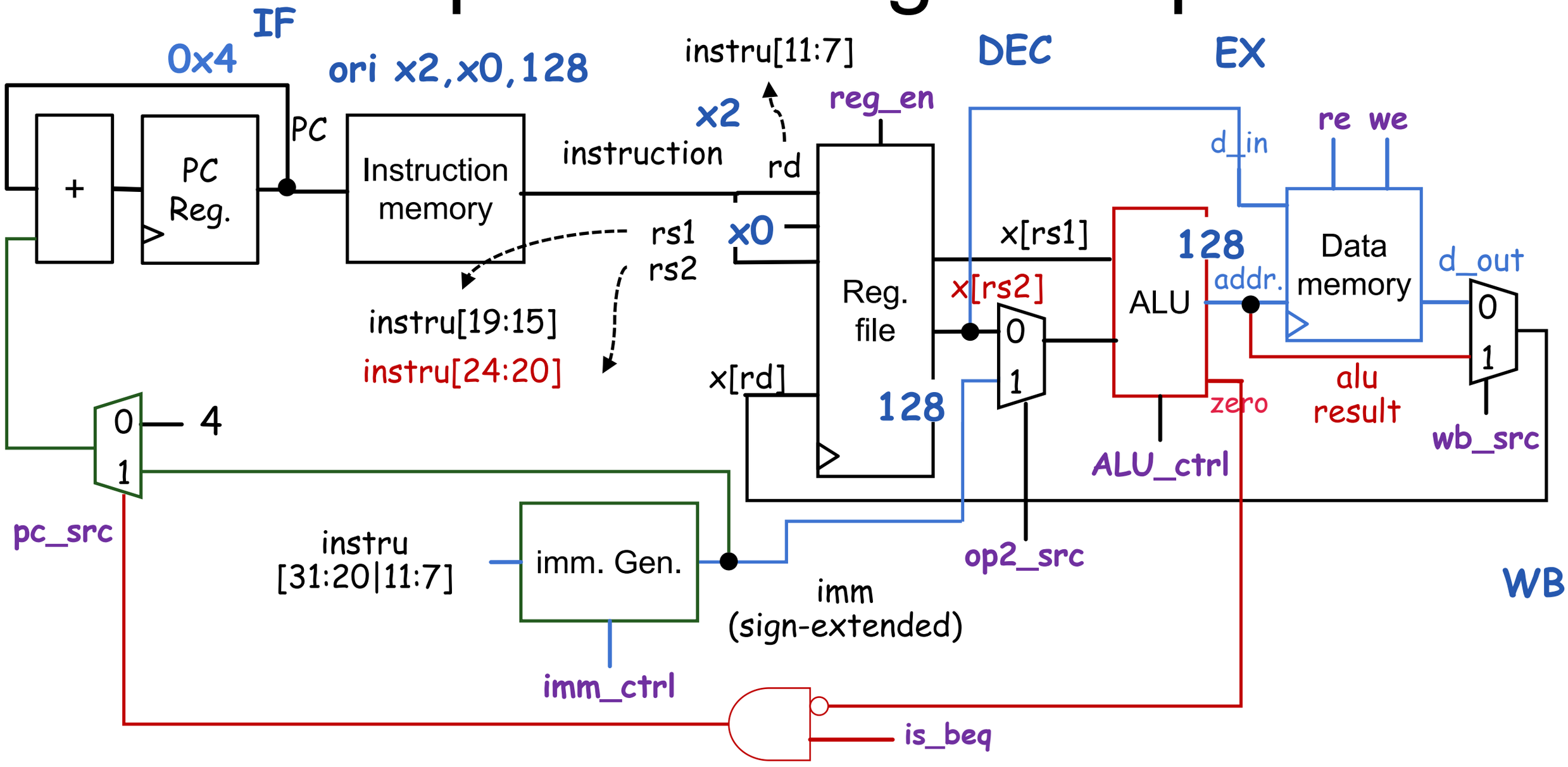
Datapath working example



	reg_en	re	we	alu_ctrl	imm_ctrl	wb_src	op2_src	is_beq
addi	1	0	0	add	I-type	1	1	0

0x0: addi x1, x0, -1
0x4: ori x2, x0, 128
0x8: add x3, x1, x2
0xc: sw x3, 0(x3)
0x10:lw x5, 0(x3)
0x14:beq x3, x5, -12

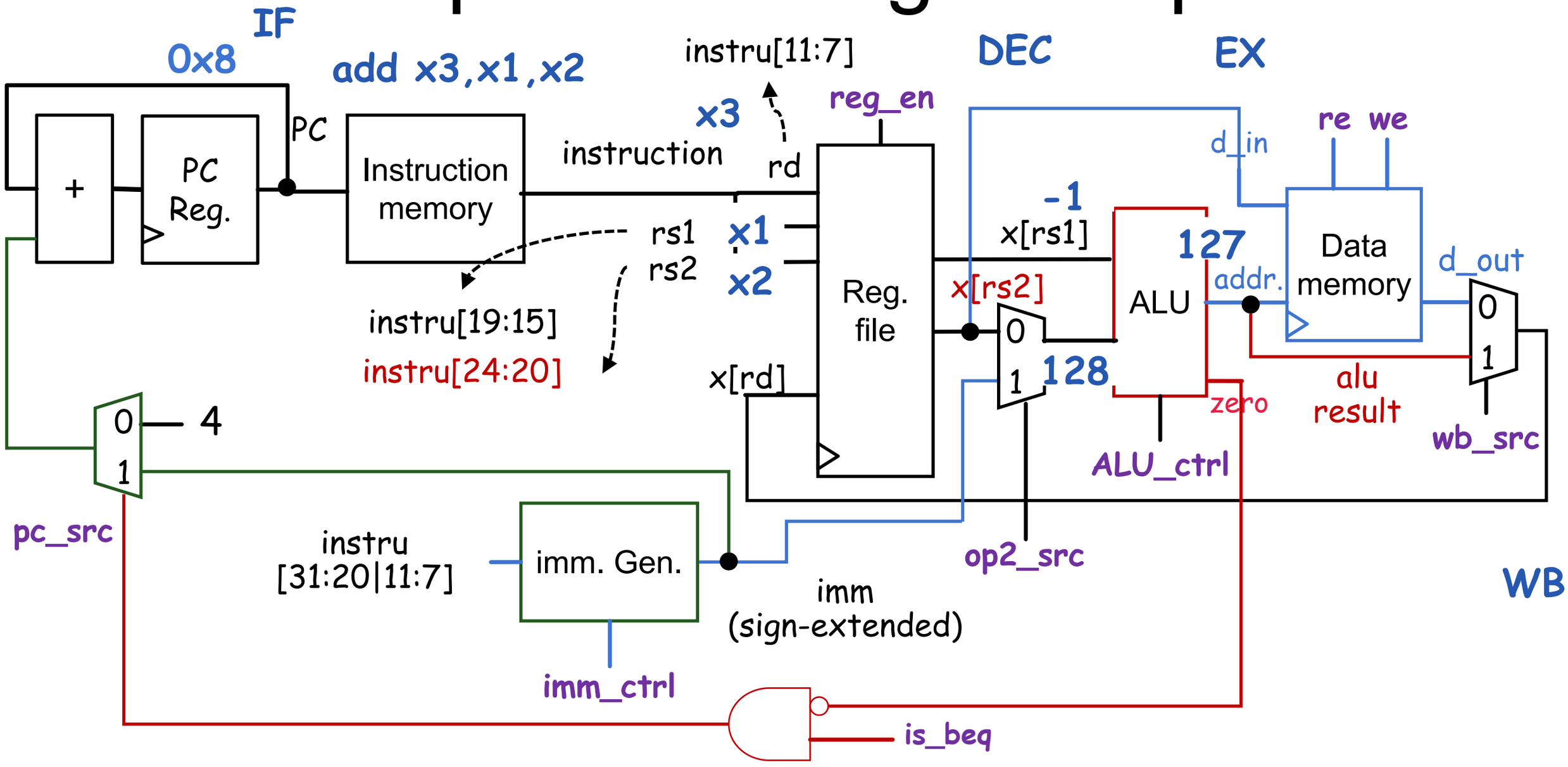
Datapath working example



	reg_en	re	we	alu_ctrl	imm_ctrl	wb_src	op2_src	is_beq
ori	1	0	0	or	I-type	1	1	0

0x0: addi x1, x0, -1
0x4: ori x2, x0, 128
0x8: add x3, x1, x2
0xc: sw x3, 0(x3)
0x10:lw x5, 0(x3)
0x14:beq x3, x5, -12

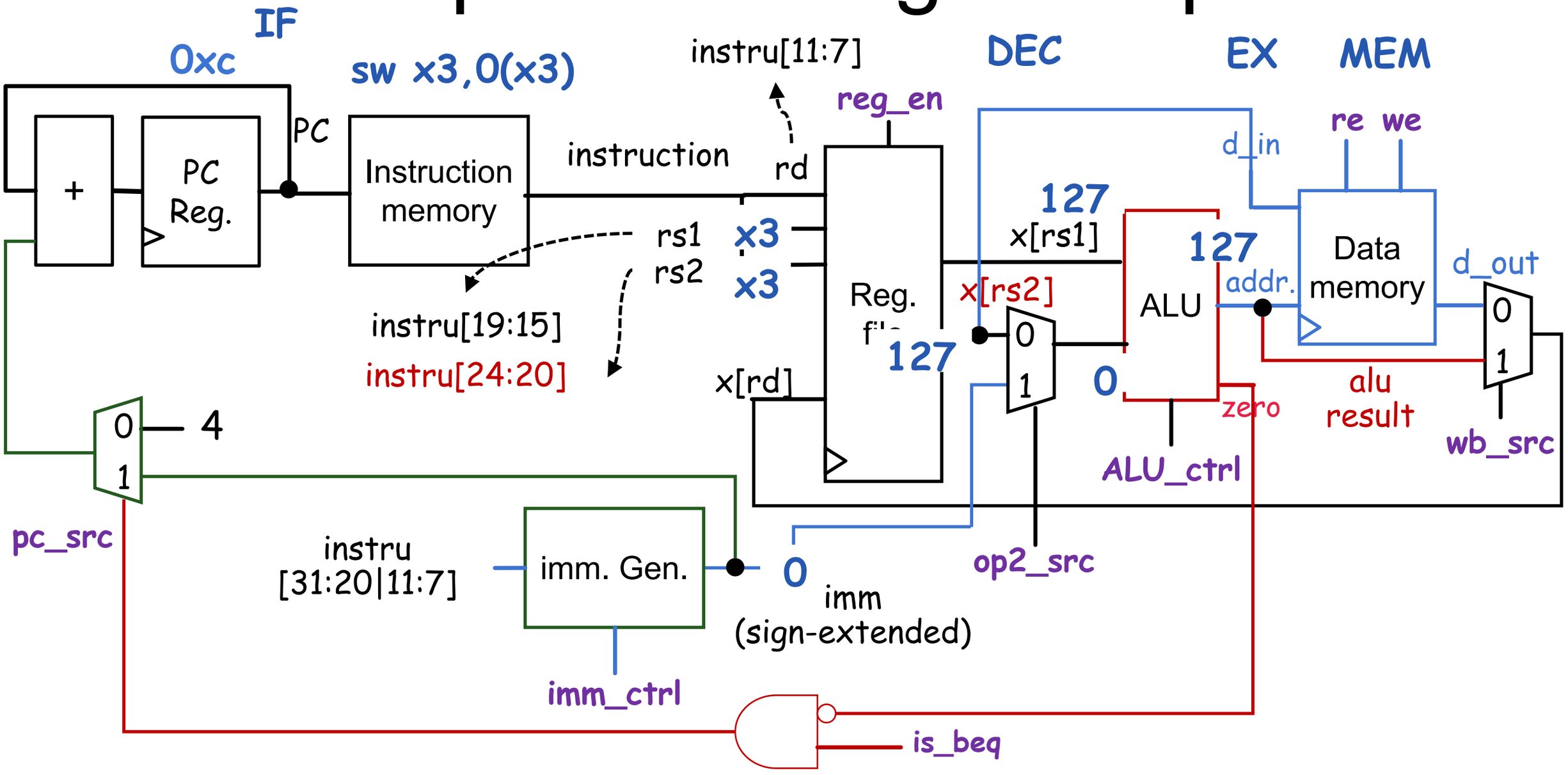
Datapath working example



	reg_en	re	we	alu_ctrl	imm_ctrl	wb_src	op2_src	is_beq
add	1	0	0	add	X	1	0	0

0x0: addi x1, x0, -1
0x4: ori x2, x0, 128
0x8: add x3, x1, x2
0xc: sw x3, 0(x3)
0x10:lw x5, 0(x3)
0x14:beq x3, x5, -12

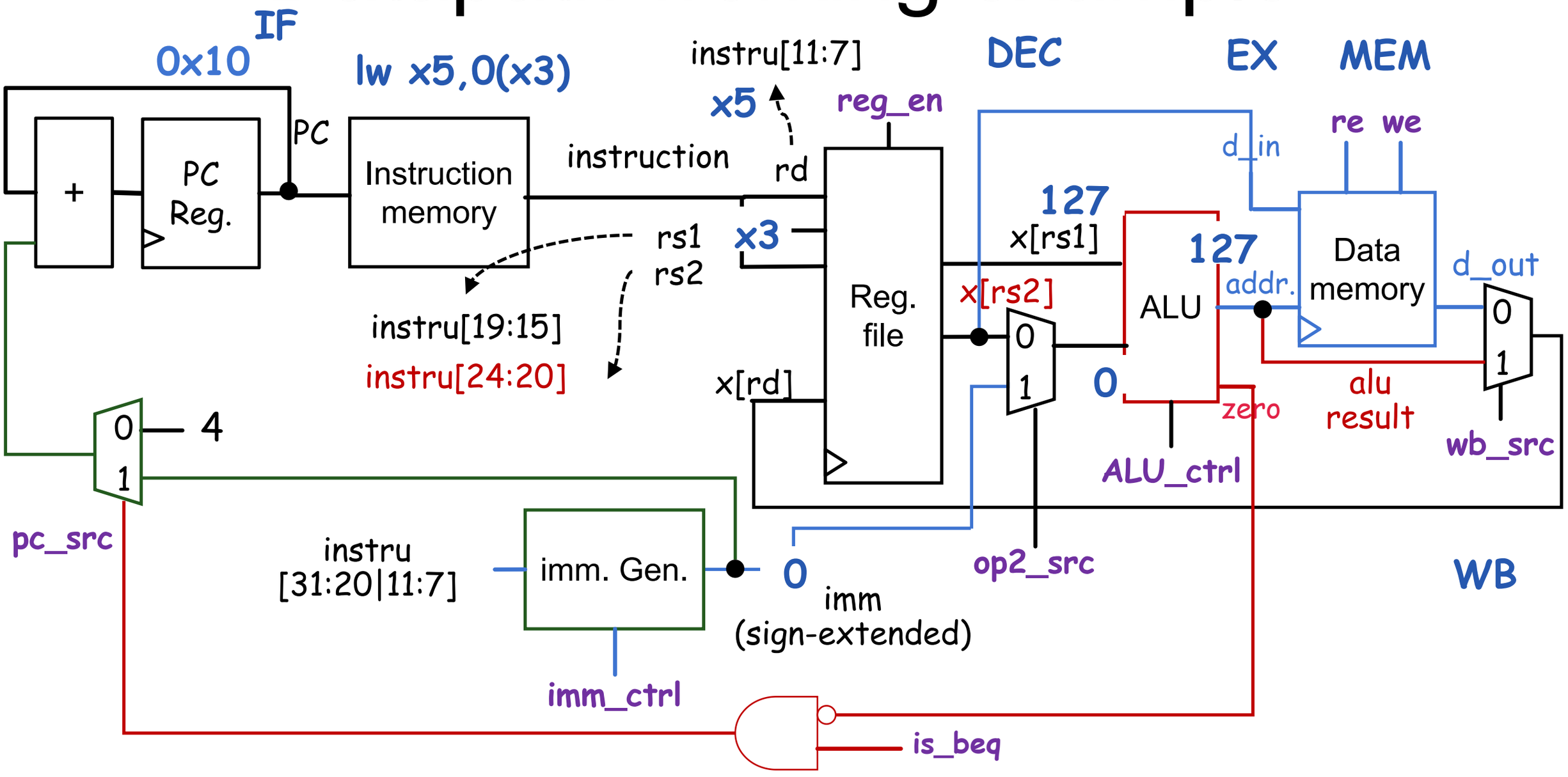
Datapath working example



	reg_en	re	we	alu_ctrl	imm_ctrl	wb_src	op2_src	is_beq
sw	0	0	1	add	S-type	X	1	0

```
0x0: addi x1, x0, -1
0x4: ori  x2, x0, 128
0x8: add  x3, x1, x2
0xc: sw   x3, 0(x3)
0x10:lw   x5, 0(x3)
0x14:beq  x3, x5, -12
```

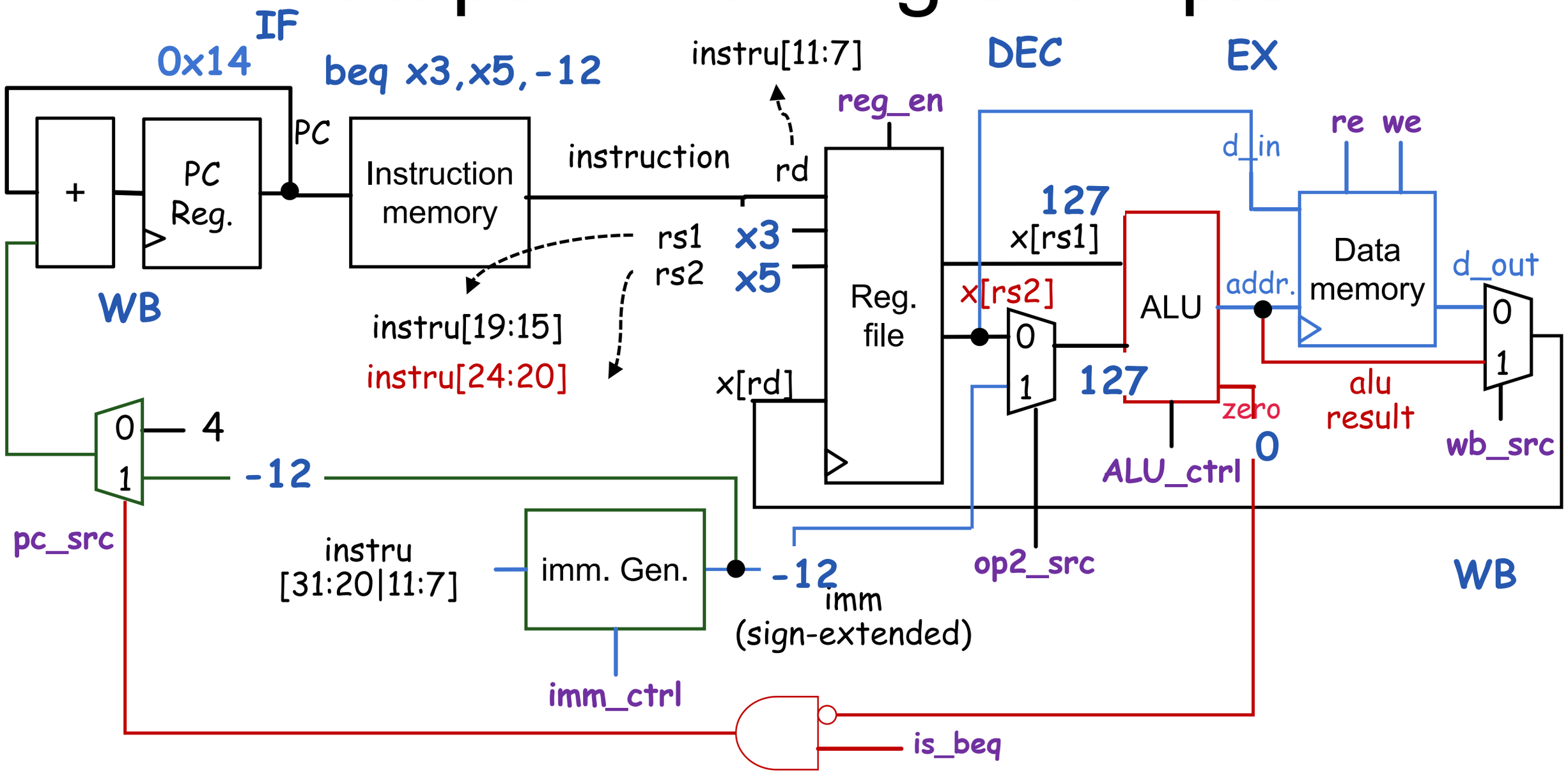
Datapath working example



	reg_en	re	we	alu_ctrl	imm_ctrl	wb_src	op2_src	is_beq
lw	1	1	0	add	I-type	0	1	0

0x0: addi x1, x0, -1
0x4: ori x2, x0, 128
0x8: add x3, x1, x2
0xc: sw x3, 0(x3)
0x10:lw x5, 0(x3)
0x14:beq x3, x5, -12

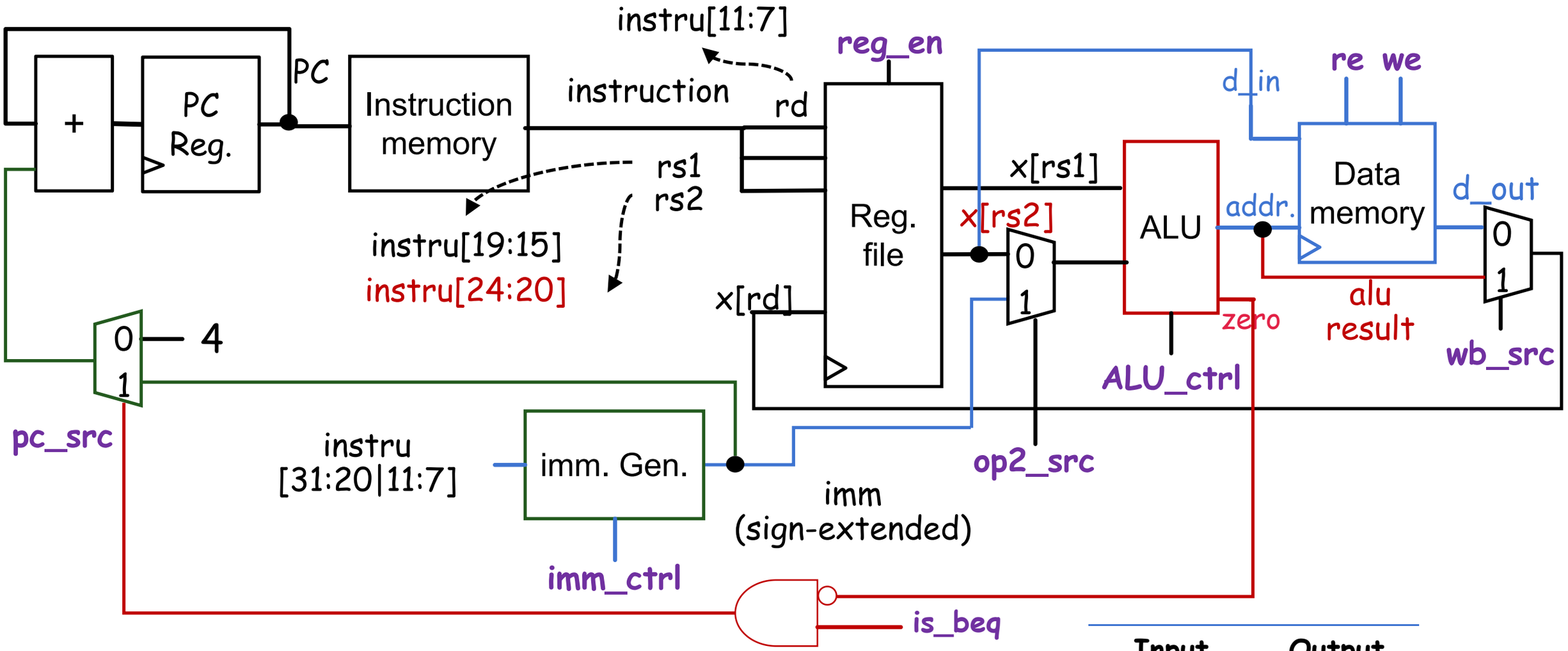
Datapath working example



	reg_en	re	we	alu_ctrl	imm_ctrl	wb_src	op2_src	is_beq
beq	0	0	0	X	B-type	X	0	1

0x0: addi x1, x0, -1
0x4: ori x2, x0, 128
0x8: add x3, x1, x2
0xc: sw x3, 0(x3)
0x10:lw x5, 0(x3)
0x14:beq x3, x5, -12

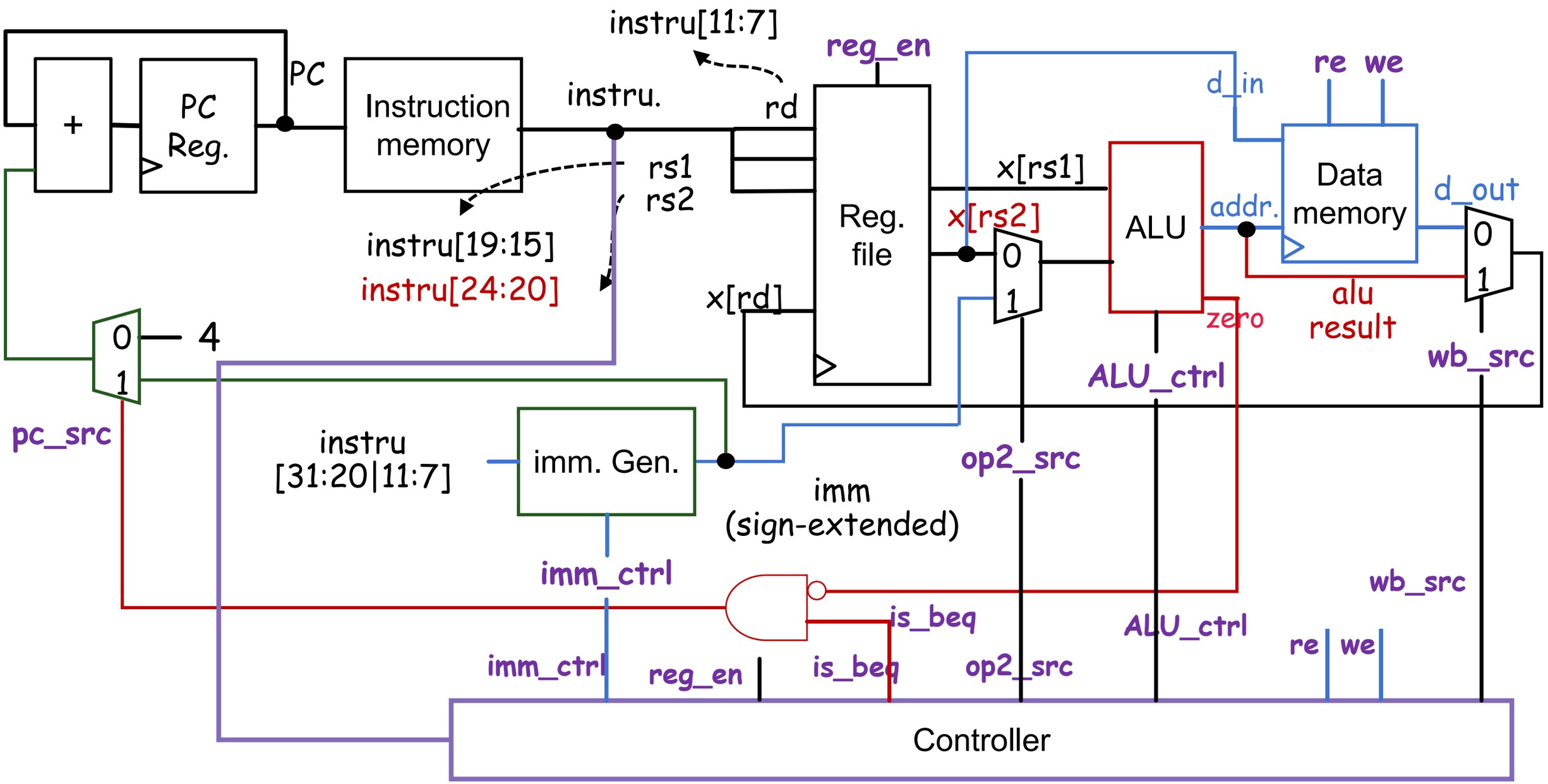
Controller



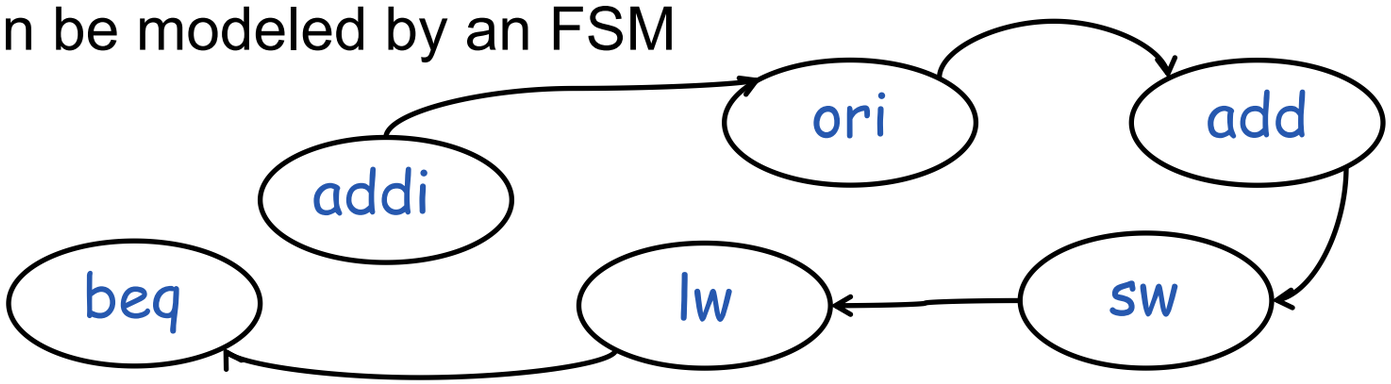
- Generate control signals guiding the datapath to execute instructions
- The inputs are instructions, the outputs are the control signals
- Once the type of instruction is determined, the control signal is determined

Input	Output
Instru.	reg_en
	re
	we
	alu_ctrl
	imm_ctrl
	wb_src
	op2_src
	is_beq

Controller

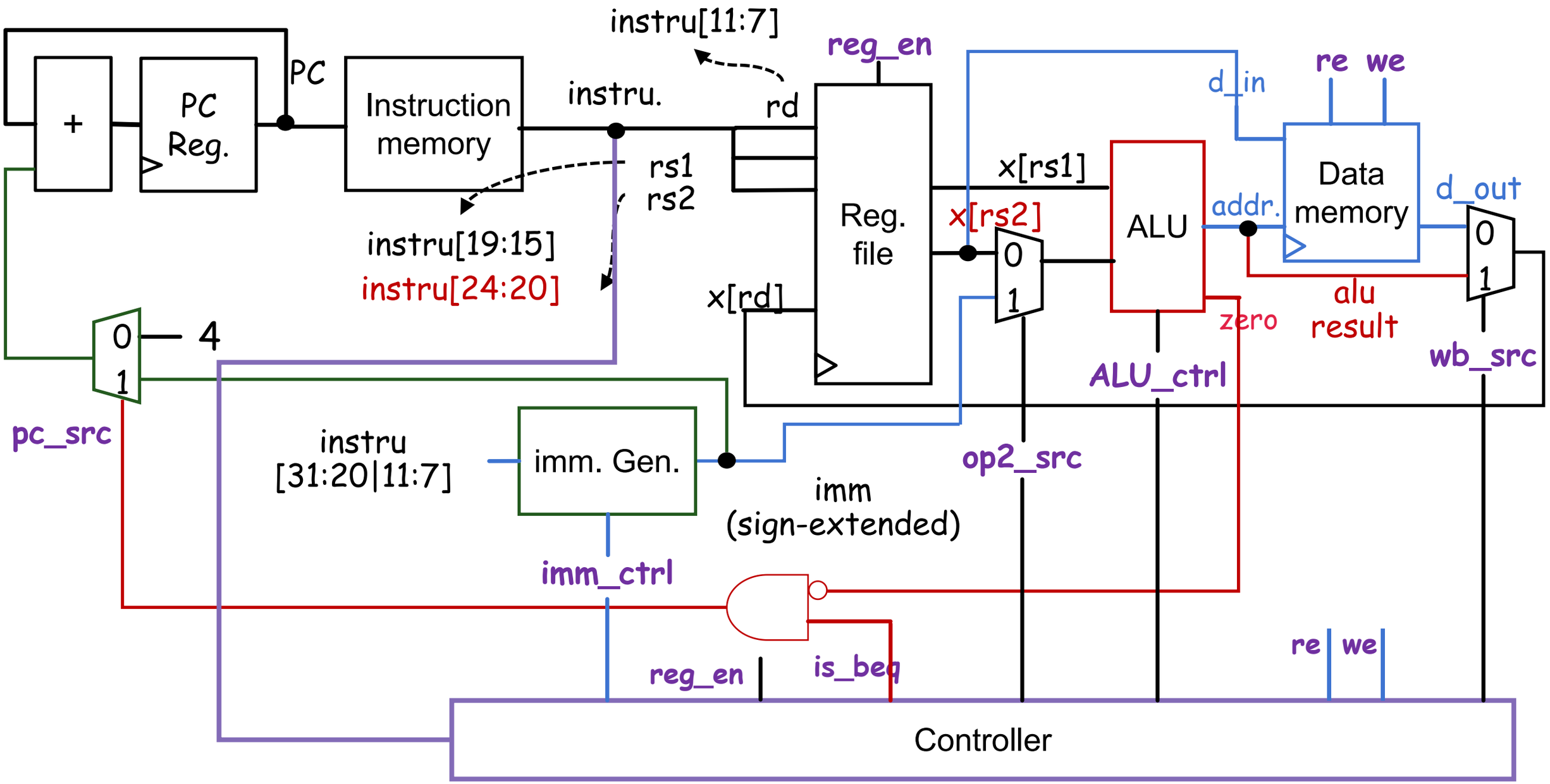


- Can be modeled by an FSM

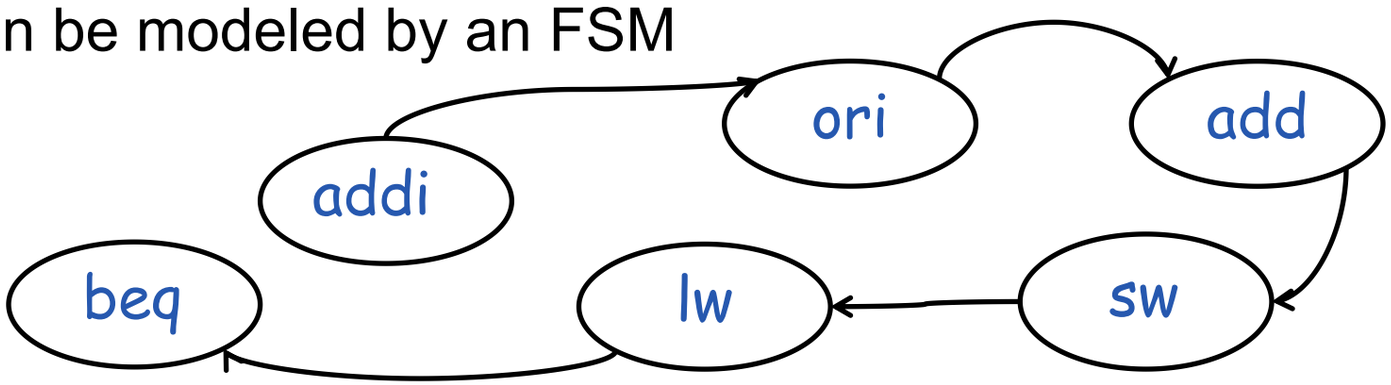



```
0x0: addi x1, x0, -1
0x4: ori  x2, x0, 128
0x8: add  x3, x1, x2
0xc: sw   x3, 0(x3)
0x10:lw   x5, 0(x3)
0x14:beq  x3, x5, -12
```

Controller

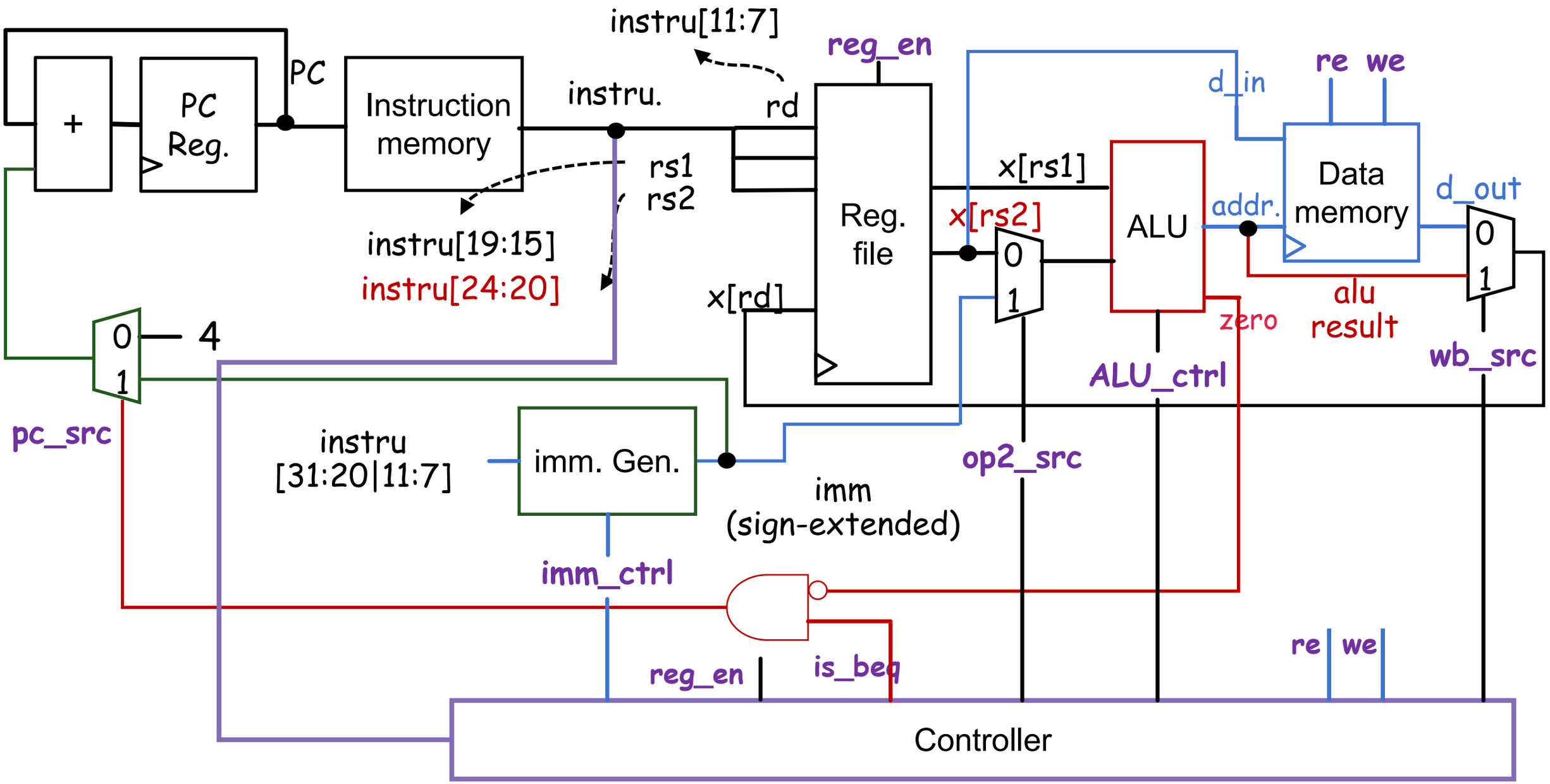


- Can be modeled by an FSM

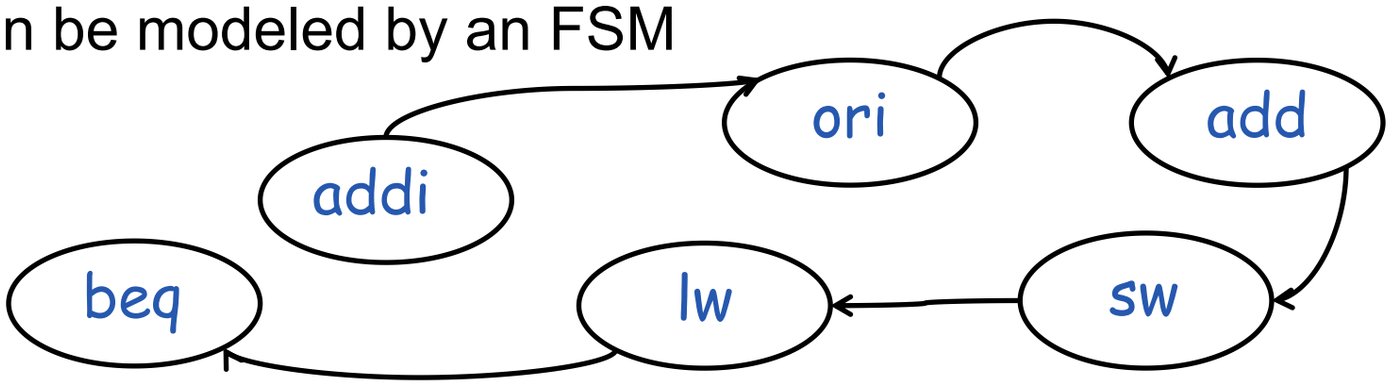


- Can be implemented by a Mealy machine

Controller

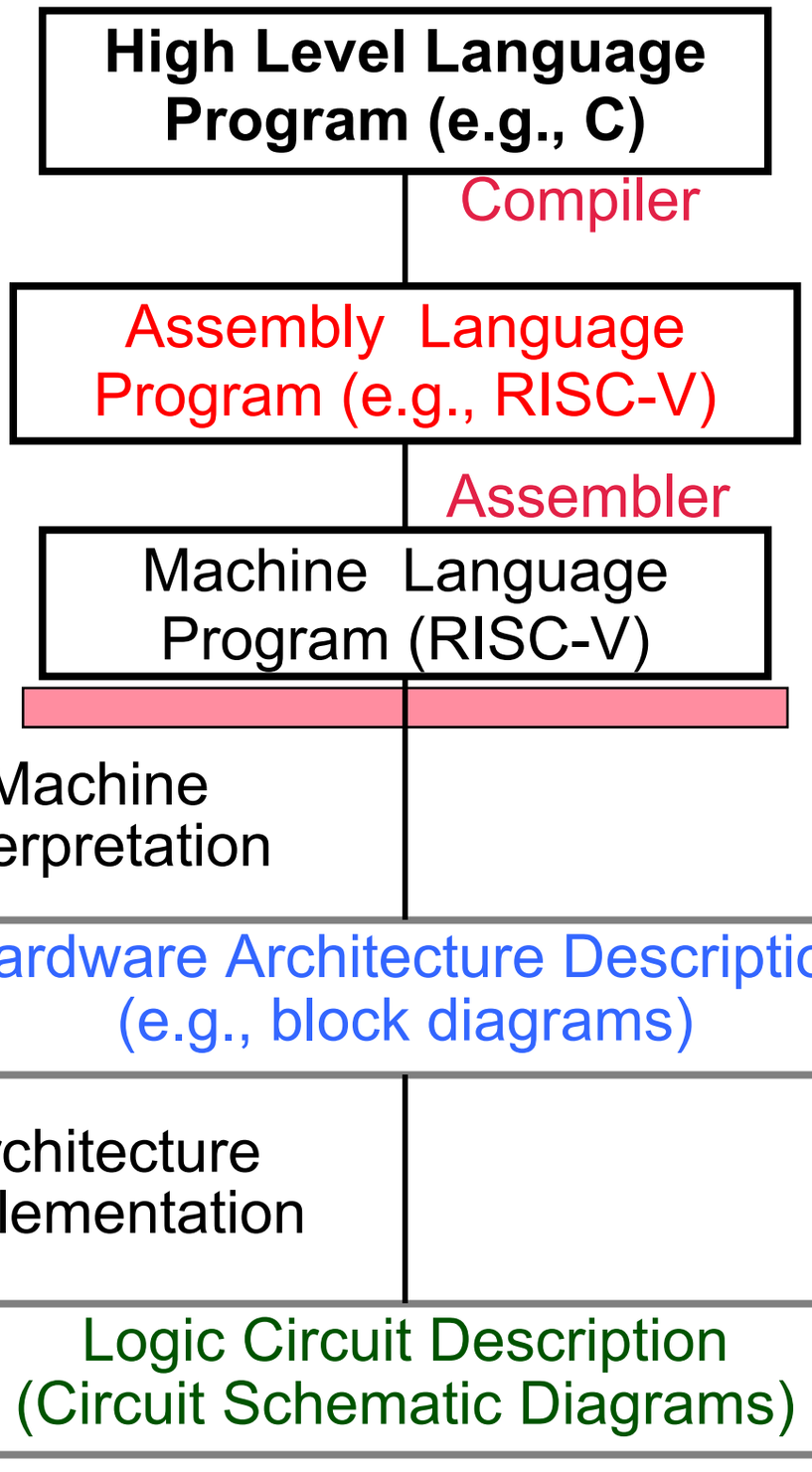


- Can be modeled by an FSM



- Can also be implemented by read-only memory (ROM), optional

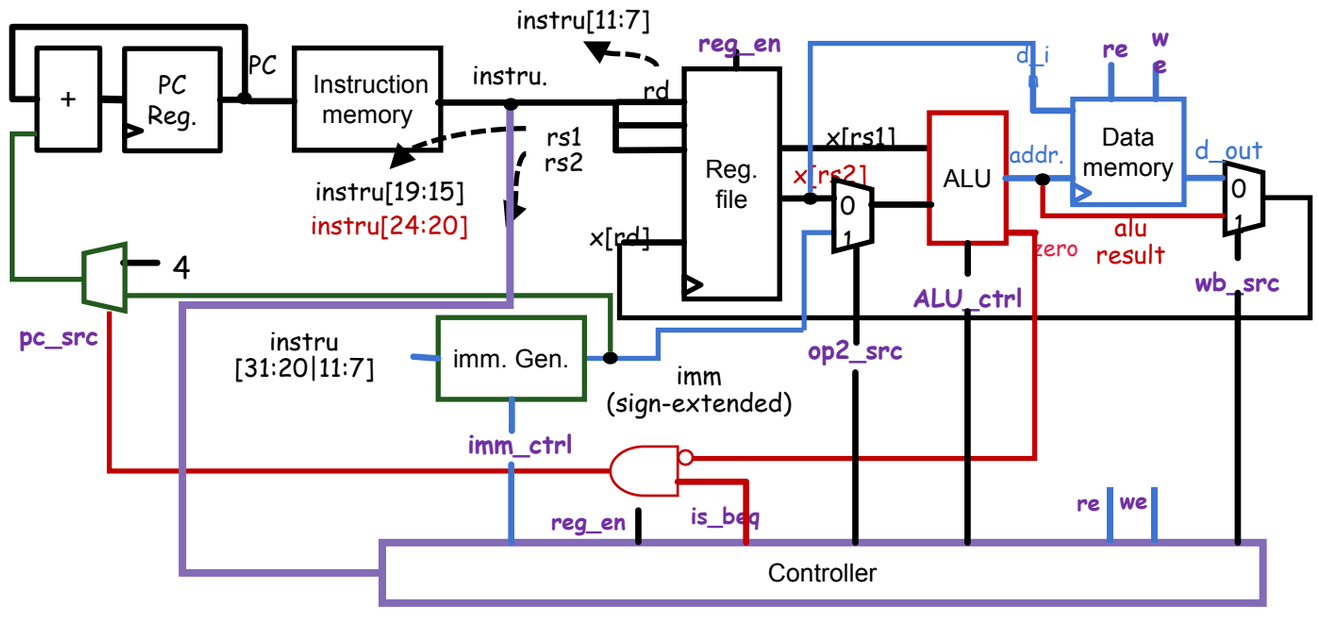
Full stack explained



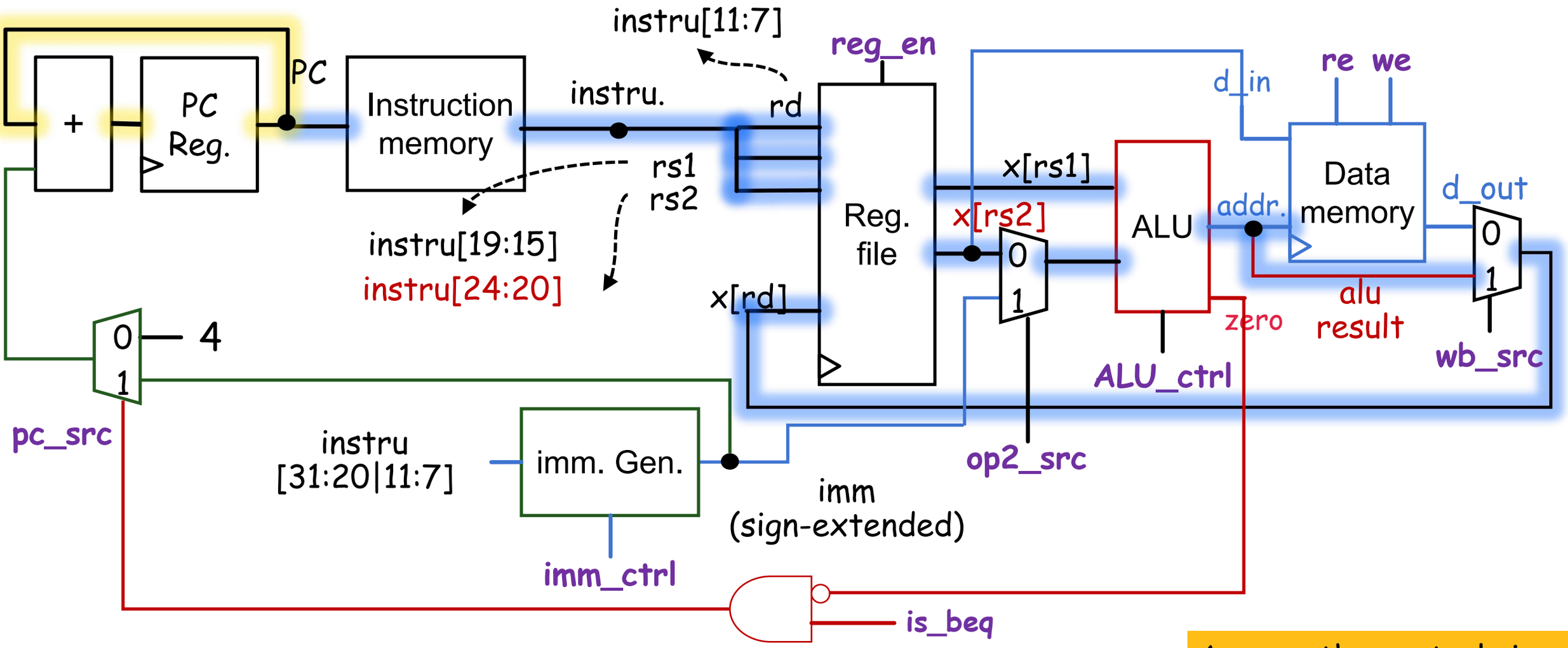

```
temp = v[k];
v[k] = v[k+1];
v[k+1] = temp;
```

```
lw    t0, 0(s2)
lw    t1, 4(s2)
sw    t1, 0(s2)
sw    t0, 4(s2)
```

```
0000 1001 1100 0110 1010 1111 0101 1000
1010 1111 0101 1000 0000 1001 1100 0110
1100 0110 1010 1111 0101 1000 0000 1001
0101 1000 0000 1001 1100 0110 1010 1111
```



Datapath timing analysis



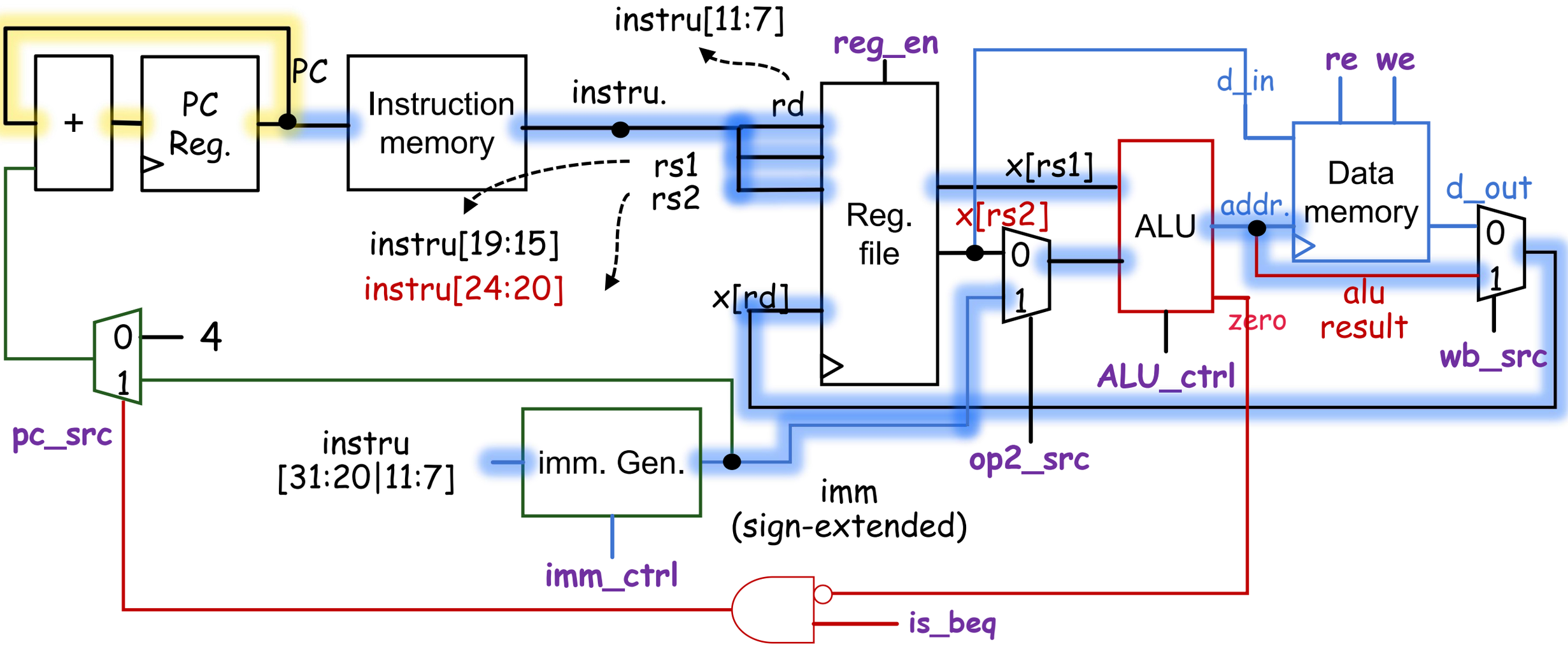
Assume the control signal is fast

$$t_{clk-to-q} + t_{add} + t_{setup}$$

$$\underbrace{t_{clk-to-q} + t_{Imem}}_{t_{IF}} + \underbrace{t_{reg} + t_{mux}}_{t_{DEC}} + \underbrace{t_{alu}}_{t_{EX}} + \underbrace{t_{mux} + t_{setup}}_{t_{WB}}$$

R-type datapath

Datapath timing analysis



$t_{clk-to-q} + t_{add} + t_{setup}$

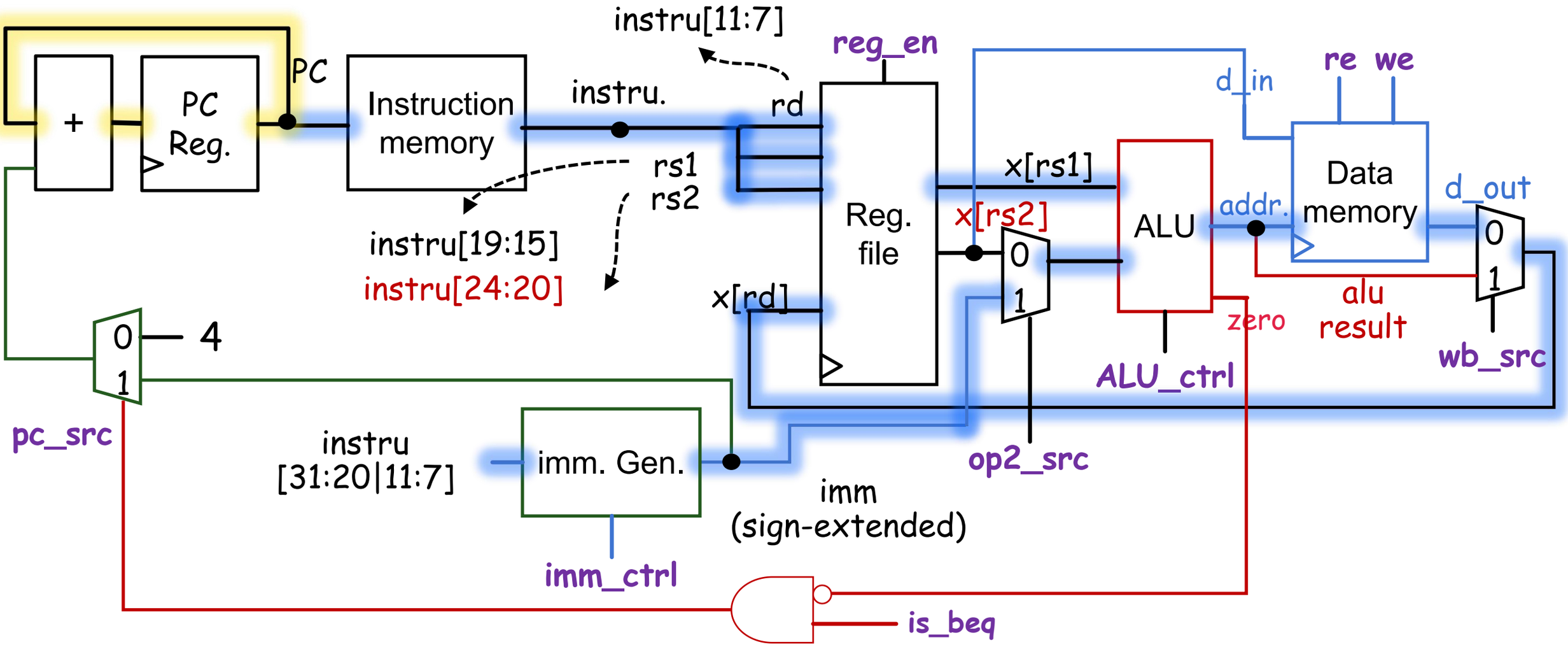
$t_{IF} + t_{DEC} + t_{EX} + t_{WB}$

$t_{clk-to-q} + t_{Imem} + t_{reg} + t_{alu} + t_{mux} + t_{setup}$

$t_{clk-to-q} + t_{Imem} + t_{imm} + t_{mux} + t_{alu} + t_{mux} + t_{setup}$ } max

I-type arithmetic & logic datapath

Datapath timing analysis



$t_{clk-to-q} + t_{add} + t_{setup}$

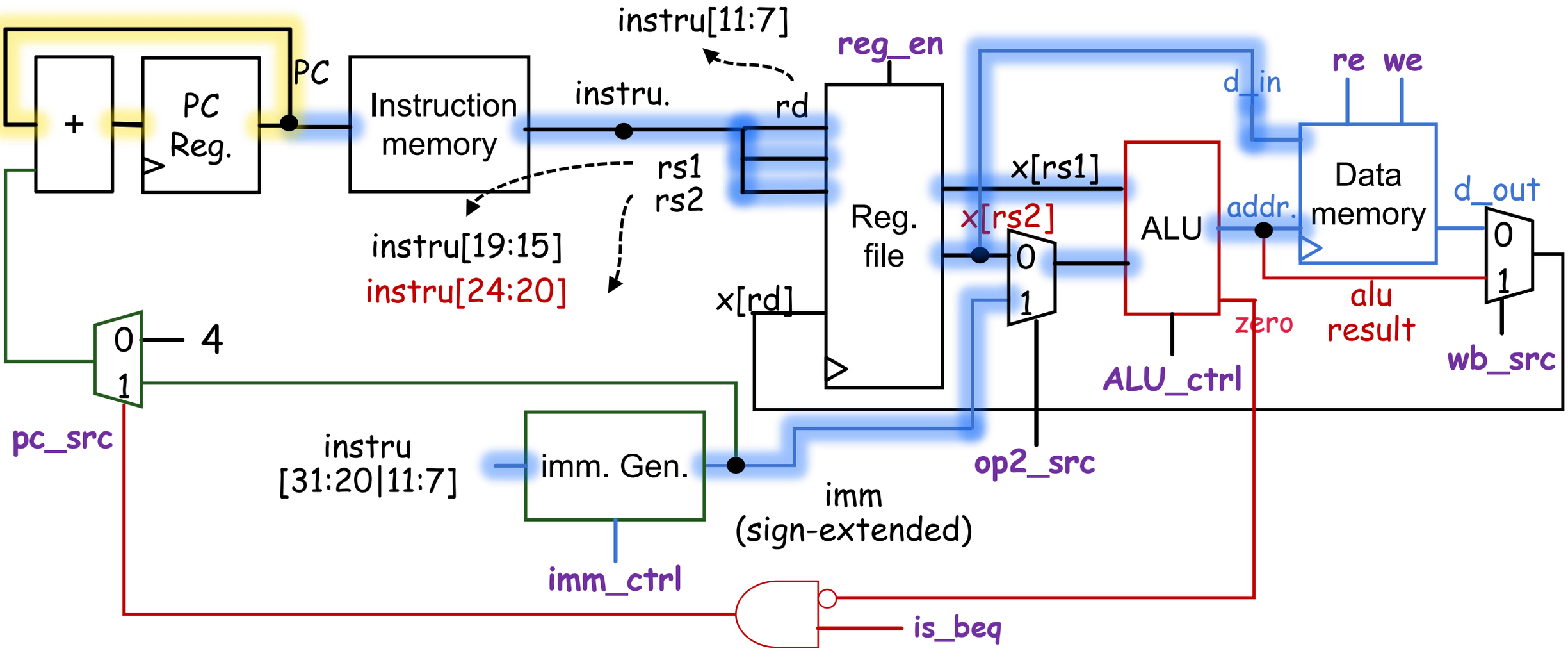
$t_{IF} + t_{DEC} + t_{EX} + t_{MEM} + t_{WB}$

$t_{clk-to-q} + t_{Imem} + t_{reg} + t_{alu} + t_{Dmem} + t_{mux} + t_{setup}$

$t_{clk-to-q} + t_{Imem} + t_{imm} + t_{mux} + t_{alu} + t_{Dmem} + t_{mux} + t_{setup}$ } max

I-type load datapath

Datapath timing analysis



$t_{clk-to-q} + t_{add} + t_{setup}$

$t_{clk-to-q} + t_{Imem} + t_{reg} + t_{Dmem}$

$t_{clk-to-q} + t_{Imem} + t_{reg} + t_{alu} + t_{Dmem}$

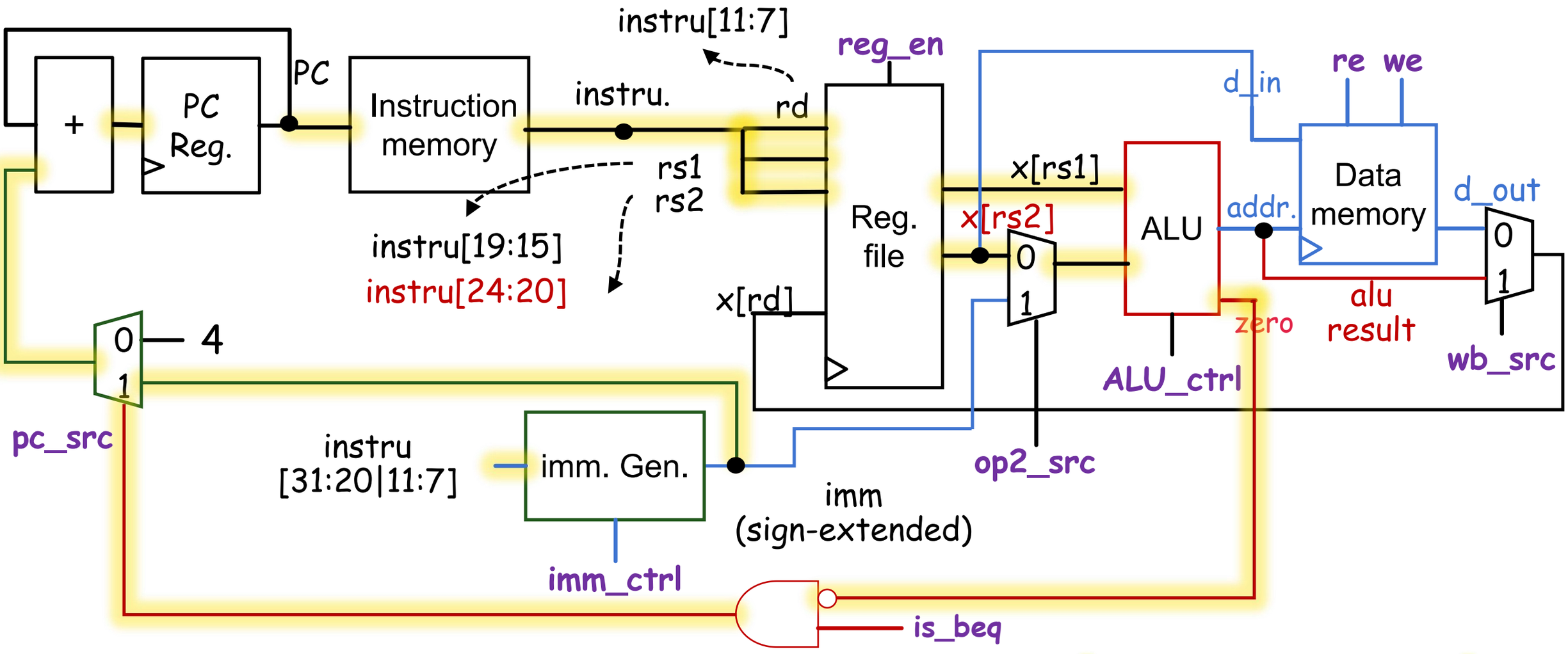
$t_{clk-to-q} + t_{Imem} + t_{imm} + t_{mux} + t_{alu} + t_{Dmem}$

} max

$t_{IF} + t_{DEC} + t_{EX} + t_{MEM}$

S-type datapath

Datapath timing analysis



$$t_{IF} + t_{DEC} + t_{EX} + t_{WB}$$

$$t_{clk-to-q} + t_{Imem} + t_{reg} + t_{mux} + t_{alu} + t_{and} + t_{mux} + t_{add} + t_{setup}$$

$$t_{clk-to-q} + t_{Imem} + t_{imm} + t_{mux} + t_{add} + t_{setup}$$

} max

B-type datapath

Summary

- We have built a single-cycle CPU
- It supports R-type, I-type arithmetic & logic and load (*lw*), S-type *sw* and *beq*
- Datapath and controller are built seperately
- Different instruction activates different parts or steps/stages (IF/DEC/EXE/MEM/WB) of the datapath, thus has different delays. The longest delay (critical path) is used to estimate maximum frequency
- Nearly no CPU uses single-cycle design today.