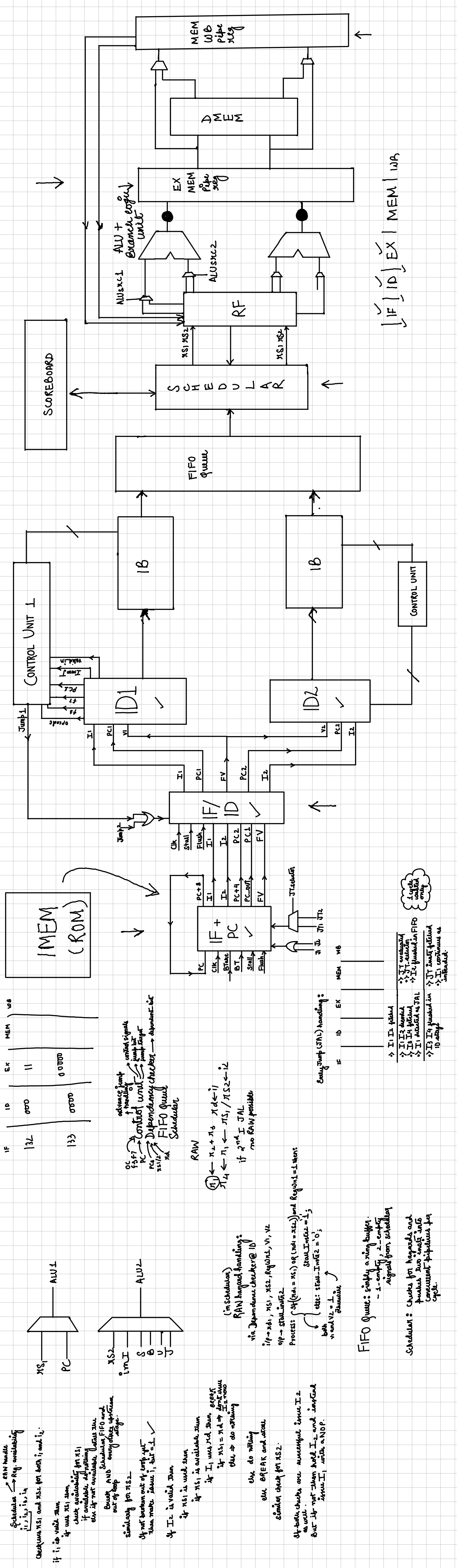
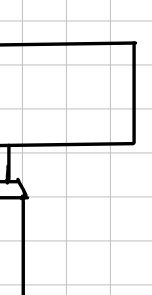
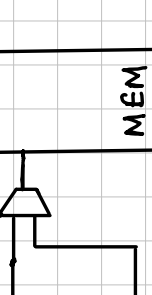
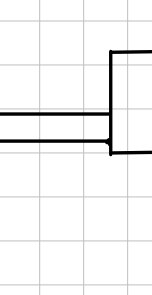
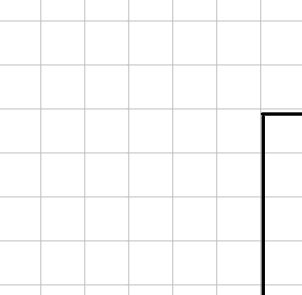
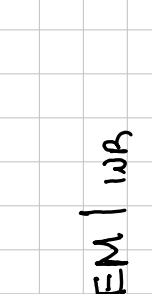
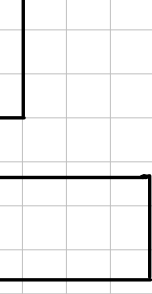
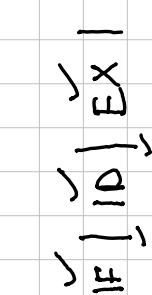
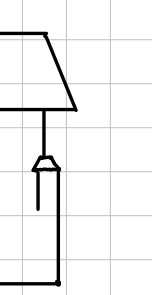
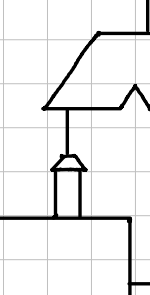
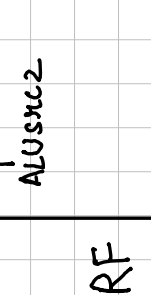
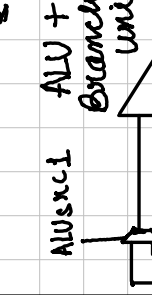
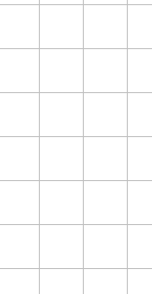
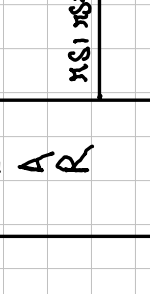
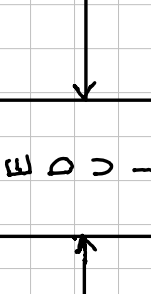
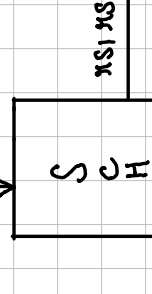
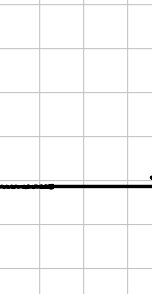
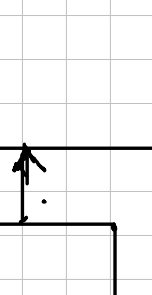
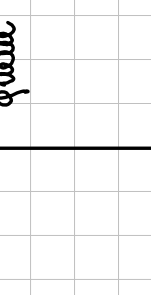
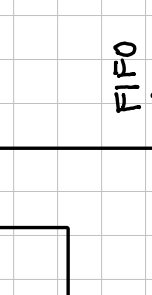
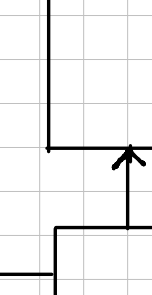
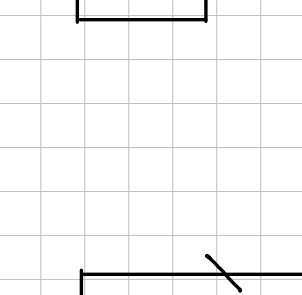
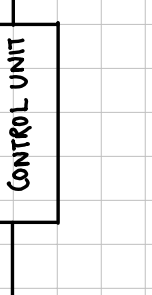
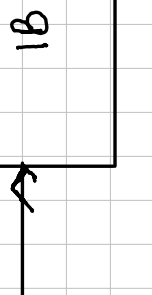
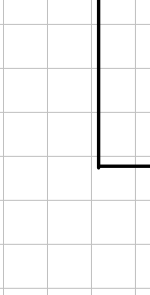
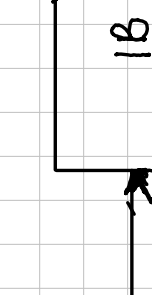
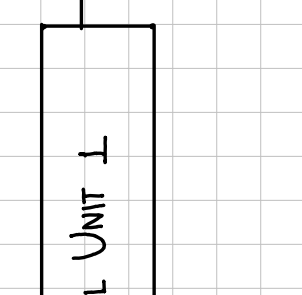
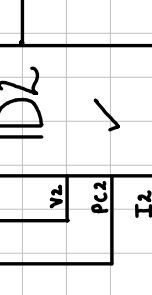
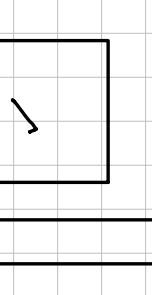
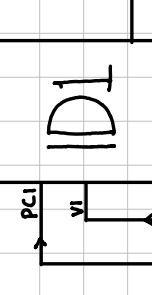
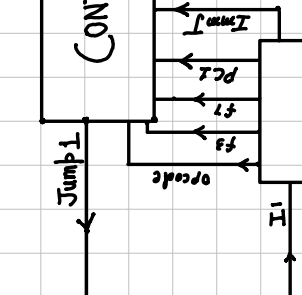
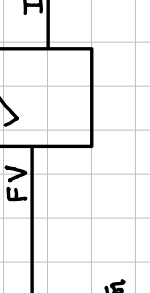
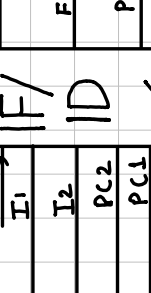
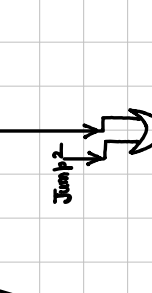
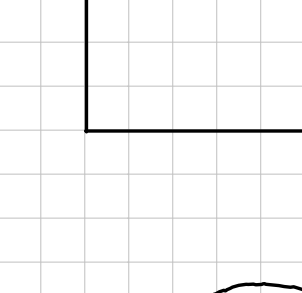
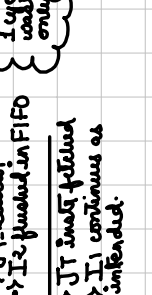
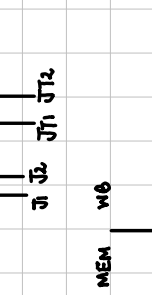
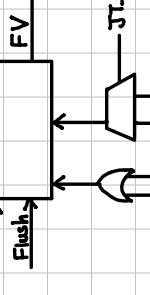
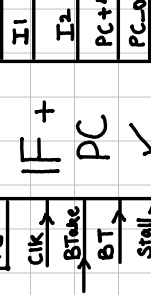
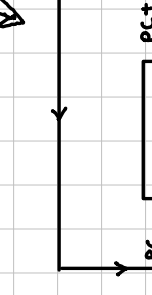
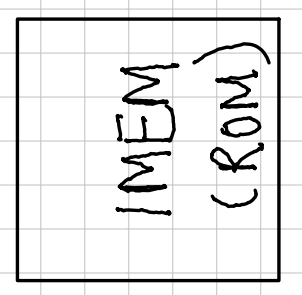
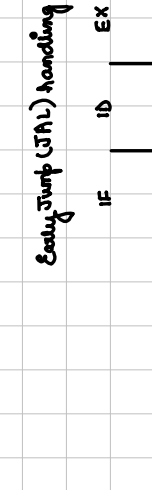
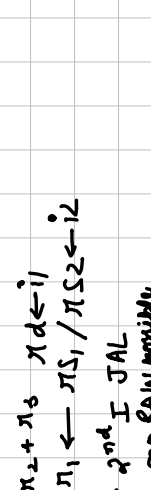
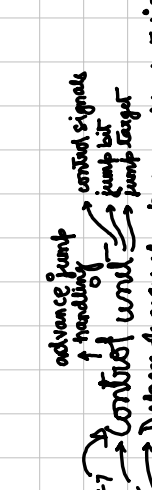
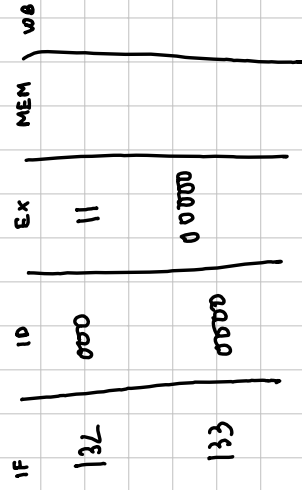


Scheduler → RAW handle
 i_1, i_2, i_3, i_4
 Checks x_{s1} and x_{s2} for both i_1 and i_2 .
 if i_1 is valid then
 if x_{s1} is available then
 check availability for x_{s2}
 if available do nothing
 else if not available (data hazard)
 Scheduler, FIFO and
 out of loop
 similarly for x_{s2}
 if not broken out of loop yet
 then make $issue_i$, but = 1 ✓

if i_2 is valid then
 if x_{s1} is valid then
 if x_{s1} is available then
 if i_1 was not then BREAK
 if $x_{s1} = x_{i1} \Rightarrow$ data hazard
 else \Rightarrow do nothing
 else do nothing
 else BREAK and stall
 similar check for x_{s2} .

if both checks are successful issue i_2
 but if not then hold i_2 and instead
 issue i_1 with a NOP.



FIFO Queue: simply a ring buffer.
 - 1 - empty, 2 - empty
 signals from scheduler

Scheduler: Checks for hazards and
 pushes two insts into
 concurrent pipelines for
 cycle.

Early Jump (JAL) handling:
 IF ID EX MEM WB
 I1 I2
 I3 I4
 I5 I6
 I7 I8
 I9 I10
 I11 I12
 I13 I14
 I15 I16
 I17 I18
 I19 I20
 I21 I22
 I23 I24
 I25 I26
 I27 I28
 I29 I30
 I31 I32
 I33 I34
 I35 I36
 I37 I38
 I39 I40
 I41 I42
 I43 I44
 I45 I46
 I47 I48
 I49 I50
 I51 I52
 I53 I54
 I55 I56
 I57 I58
 I59 I60
 I61 I62
 I63 I64
 I65 I66
 I67 I68
 I69 I70
 I71 I72
 I73 I74
 I75 I76
 I77 I78
 I79 I80
 I81 I82
 I83 I84
 I85 I86
 I87 I88
 I89 I90
 I91 I92
 I93 I94
 I95 I96
 I97 I98
 I99 I100

Early Jump (JAL) handling:
 IF ID EX MEM WB
 I1 I2
 I3 I4
 I5 I6
 I7 I8
 I9 I10
 I11 I12
 I13 I14
 I15 I16
 I17 I18
 I19 I20
 I21 I22
 I23 I24
 I25 I26
 I27 I28
 I29 I30
 I31 I32
 I33 I34
 I35 I36
 I37 I38
 I39 I40
 I41 I42
 I43 I44
 I45 I46
 I47 I48
 I49 I50
 I51 I52
 I53 I54
 I55 I56
 I57 I58
 I59 I60
 I61 I62
 I63 I64
 I65 I66
 I67 I68
 I69 I70
 I71 I72
 I73 I74
 I75 I76
 I77 I78
 I79 I80
 I81 I82
 I83 I84
 I85 I86
 I87 I88
 I89 I90
 I91 I92
 I93 I94
 I95 I96
 I97 I98
 I99 I100

IF ID EX MEM WB
 I1 I2
 I3 I4
 I5 I6
 I7 I8
 I9 I10
 I11 I12
 I13 I14
 I15 I16
 I17 I18
 I19 I20
 I21 I22
 I23 I24
 I25 I26
 I27 I28
 I29 I30
 I31 I32
 I33 I34
 I35 I36
 I37 I38
 I39 I40
 I41 I42
 I43 I44
 I45 I46
 I47 I48
 I49 I50
 I51 I52
 I53 I54
 I55 I56
 I57 I58
 I59 I60
 I61 I62
 I63 I64
 I65 I66
 I67 I68
 I69 I70
 I71 I72
 I73 I74
 I75 I76
 I77 I78
 I79 I80
 I81 I82
 I83 I84
 I85 I86
 I87 I88
 I89 I90
 I91 I92
 I93 I94
 I95 I96
 I97 I98
 I99 I100

IF ID EX MEM WB
 I1 I2
 I3 I4
 I5 I6
 I7 I8
 I9 I10
 I11 I12
 I13 I14
 I15 I16
 I17 I18
 I19 I20
 I21 I22
 I23 I24
 I25 I26
 I27 I28
 I29 I30
 I31 I32
 I33 I34
 I35 I36
 I37 I38
 I39 I40
 I41 I42
 I43 I44
 I45 I46
 I47 I48
 I49 I50
 I51 I52
 I53 I54
 I55 I56
 I57 I58
 I59 I60
 I61 I62
 I63 I64
 I65 I66
 I67 I68
 I69 I70
 I71 I72
 I73 I74
 I75 I76
 I77 I78
 I79 I80
 I81 I82
 I83 I84
 I85 I86
 I87 I88
 I89 I90
 I91 I92
 I93 I94
 I95 I96
 I97 I98
 I99 I100