University of California, Berkeley
College of Engineering
Department of Electrical Engineering and Computer Science

Spring 2000 Prof. Bob Brodersen

Midterm 1

March 15, 2000
CS152: Computer Architecture

This midterm consists of four problems, each of which has multiple parts, so budget your time accordingly. The exam is closed-book, but calculators and one sheet of notes are allowed. Good luck!

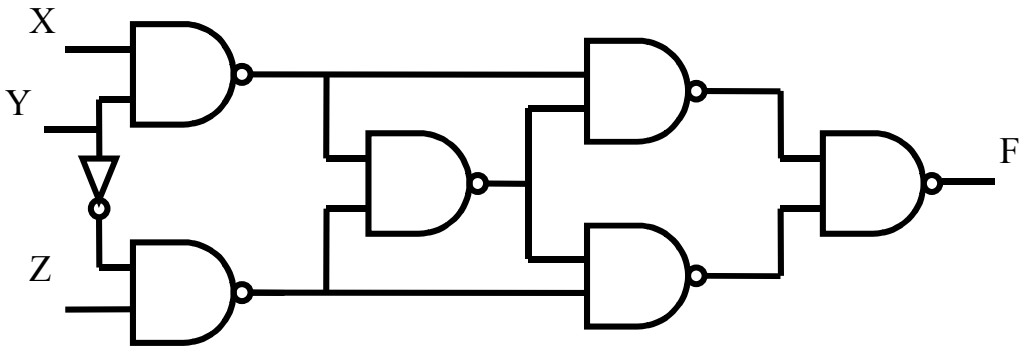
Name	SOLUTIONS
SID	
Discussion	

1	
2	
3	
4	
Total	

Problem 1: Critical Path and Delay (25 points)

Throughout this problem, use the simple linear delay model presented in class. For the circuit below, assume the following delay parameters:

NAND: t_{plh} = 0.5ns, t_{phl} = 0.5ns, t_{plhf} = 0.002ns/fF, t_{phlf} = 0.002ns/fF
Input capacitance: 100fF
Inverter: t_{plh} = 0.2ns, t_{phl} = 0.2ns, t_{plhf} = 0.001ns/fF, t_{phlf} = 0.001ns/fF
Input capacitance: 50fF
Wiring Capacitance: (Equal for all nodes) 5fF



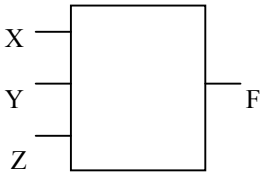
a) What is the worst case delay? Assume there is no delay at the inputs X, Y and Z.

The equation for the worst case delay is as follows:

	.2ns+105fF*.001ns/fF	(INVERTER)
	.5ns+205fF*.002ns/fF	(NAND1)
	.5ns+205fF*.002ns/fF	(NAND2)
	.5ns+105fF*.002ns/fF	(NAND3
+	.5ns+5fF*.002ns/fF	(NAND4)
	= 3.345 ns	

Note: There is no delay at the input nodes, and remember to include fan-out and wiring delay!

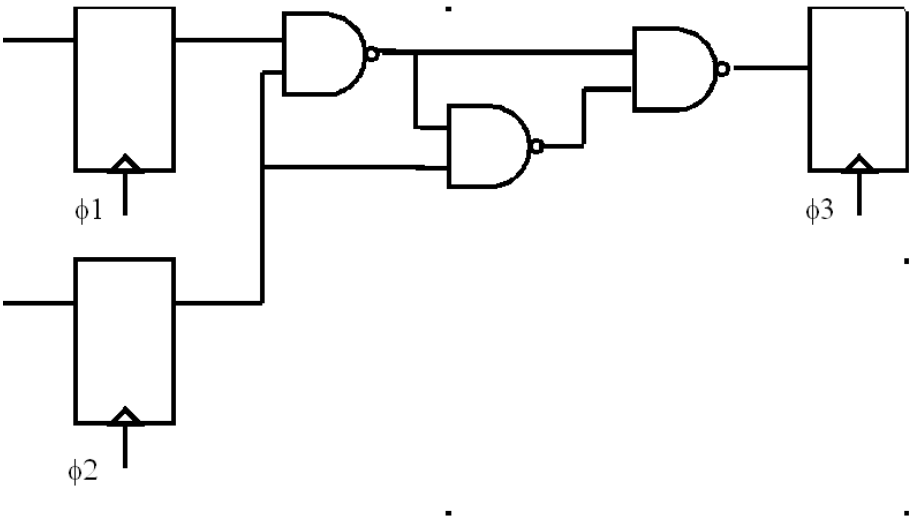
b) Now assume that you want to generate a symbol for the circuit in part (a). Determine the following parameters for your symbol: t_{plh}, t_{phl}, and the load dependant delay (in ns/fF).



First the propagation delays: t_{plh} = t_{phl}, = 3.345 ns. This is the same as the critical path from the last part. For the load dependent delay, since we only have a single NAND driving the output, it is the same as the NAND itself: 0.002 ns/fF.

Now consider the following circuit and the following parameters:

Register: $t_{clk\text{-}to\text{-}Q} = 0.6ns$, $t_{setup} = 0.5ns$, $t_{hold} = 0.2ns$
NAND: $t_{plh} = 0.5ns$, $t_{phl} = 0.5ns$



c) What is the maximum frequency at which this circuit will operate correctly? Ignore any load dependent delay.

The critical path is from either register through all three NANDs, which corresponds to the clock-to-Q of the first registers plus the propagation delay of the three NANDs plus the *setup* time of the last register.
Therefore, $F_{max} = 1/(0.6ns+1.5ns+0.5ns) = 385$ MHz

d) If the clock signals are skewed so that ϕ_1 arrives 0.2ns before ϕ_2 which arrives 0.2ns before ϕ_3 (such that ϕ_1-$\phi_2 = \phi_2$-$\phi_3 = 0.2ns$), what is the maximum frequency at which this circuit will operate correctly?

The key is the direction of the skew. In this case, since ϕ_2 comes after ϕ_1, the critical path needs to wait for ϕ_2. However, since the skew is set up such that the clock reaches the inputs earlier than the output register, we can actually *increase* our F_{max}, since the skew buys us more time. (Draw the timing diagram to prove for yourself.)
Therefore, $F_{max} = 1/(0.6ns+1.5ns+0.5ns-0.2ns) = 416$ MHz

e) Now assume that ϕ_1 and ϕ_2 arrive at the same time, and that ϕ_3 arrives later. What is the maximum tolerable skew to ensure that there are no hold time violations?

We must have clock-to-Q + shortest delay > skew + t_{hold}
To prevent hold time violations, we need to make sure that the data on the output of register 3 is stable for at least 0.2ns. This means that if data from registers 1 or 2 changes at the early clock edge, we have to guarantee that no matter what values occur, the output of register 3 does not change, thus the skew needs to be less than *1.4ns*.

Problem 2: Single-cycle Processors (25 points)

The following MIPS code finds the maximum integer within a bounded array, where $4 contains a pointer to the beginning of the array, $5 contains the length of the array and $3 contains the pointer to store the result at the end. (Assuming there is no branch delay slot.)

```
      LW $2, 0($4)  // assume the first number is the largest
      ADDI $4, $4, 4

      ADDI $5, $5, -1

max:
      LW $6, 0($4) // load array element and increment pointer
      ADDI $4, $4, 4

      SLT $7, $2, $6 // update $2 if $6 is larger
      BEQ $7, $0, next
      ADD $2, $0, $6

next:
      ADDI $5, $5, -1 // continue the search until end of array
      BEQ $5, $0, finish
      J max

finish:     SW $2, 0($3) // store result
```

The single-cycle datapath and control unit are shown on the next page. Assume that the delay and energy consumption per operation for each functional unit is as follows:

- Memory (read or write): 3 ns, 3 pJ
- ALU and adder: 2 ns, 2 pJ
- Register file (read or write): 1 ns, 1 pJ
- All other units: 0 ns, 0 pJ

a) What is the minimum clock cycle time for this processor?

The minimum cycle time of the processor is 10ns (or a frequency of 100MHz).

b) For an array of length N, what is the range of execution time for this program (e.g. – the minimum possible execution time and the worst case execution time)?

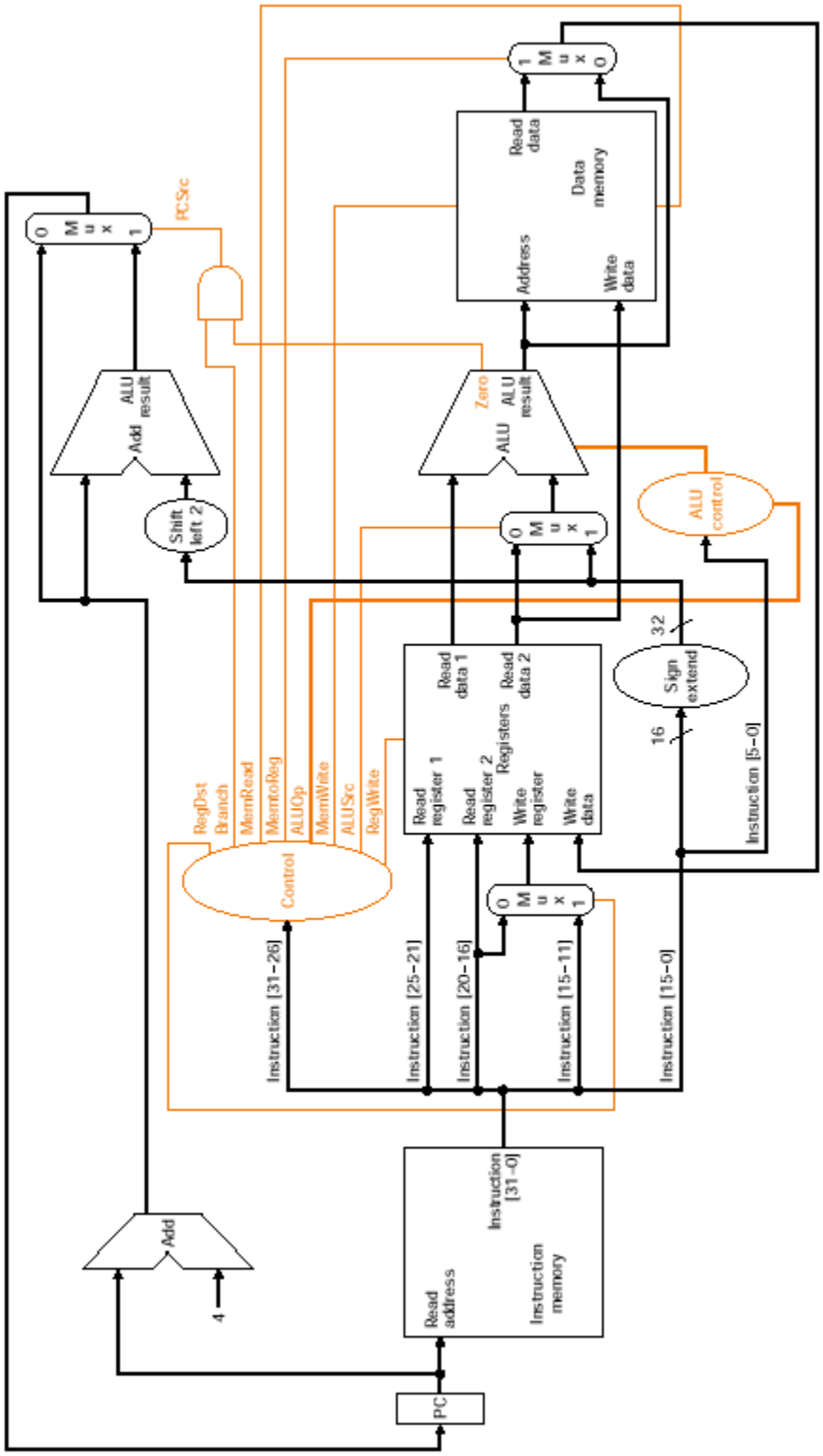
We asked for a range on this problem since the exact number of instructions is data dependent (a result of the first branch). The range of execution times is a minimum of 10(7N-4) ns and a maximum of 10(8N-5) ns.

c) What is the energy consumption (per instruction) for each type of instruction in the program? Assume that components are completely "turned off" and do not consume energy when they are not needed.

The energy consumption of each type of instruction is listed below.

LW: 12 pJ
ADDI/ADD/SLT: 9 pJ
BEQ: 10 pJ
SW: 11 pJ

Diagram and scratch space for Problem 2:



Problem 3: Single-cycle Datapath Design (25 points)

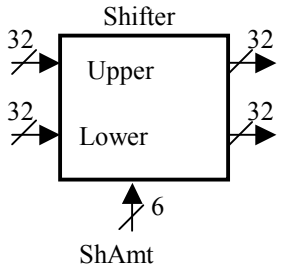
The task is to design a single cycle processor with the minimum number of functional units that can perform the following standard MIPS instructions plus a new rotate instruction. The rotate instruction does a rotate of $RS to the right by the IMMED value and stores it in $RD (e.g. a 2 shift rotate of a 5 byte word would turn [a b c d e] into [d e a b c]). The blocks that you can use are given below along with their control signals and delay values. All blocks are similar to those we used in class, except for the addition of a 64 bit shifter.

Instructions:

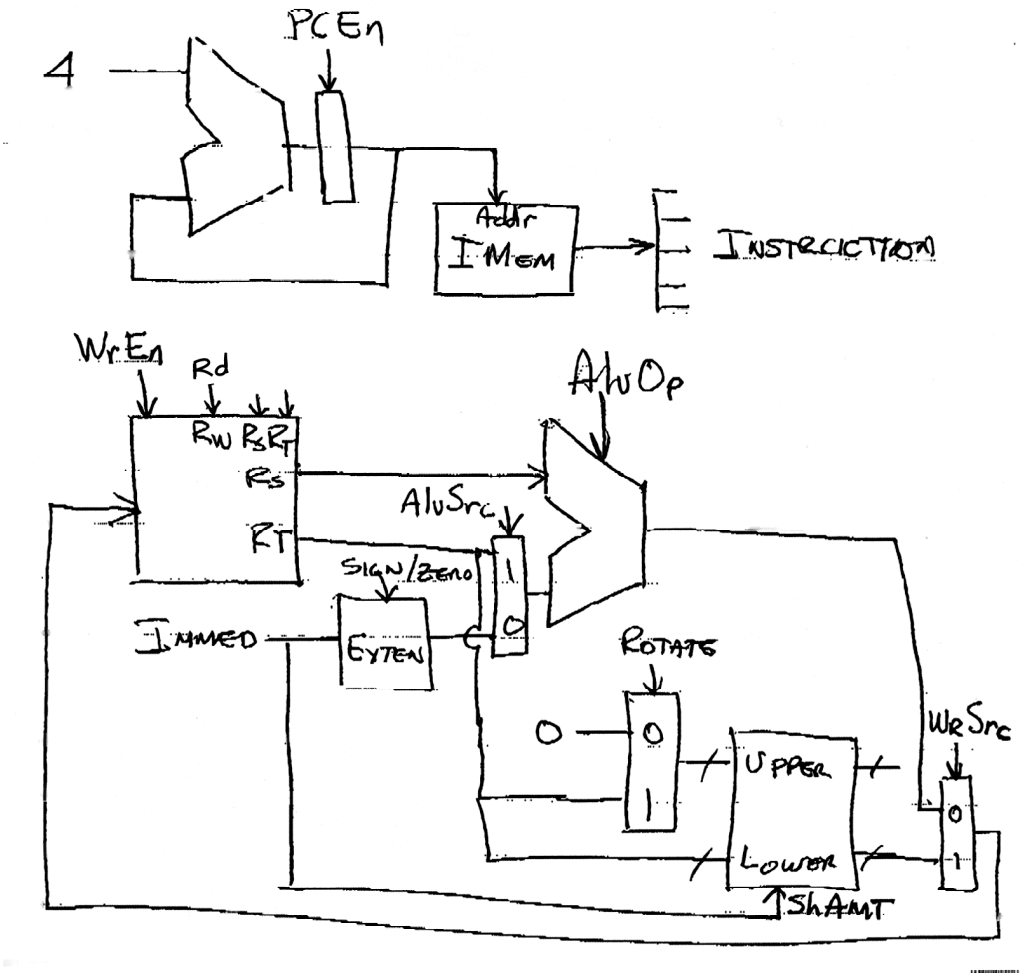
- ADDIU $RD $RS IMMED
- ADD $RD $RS $RT
- SRL $RD $RS IMMED
- Rotate $RD $RS IMMED

Components (and control signals):

- **ALU (ALUcontrol, Zero) => 32 bit ALU with Zero status bit**
  – ALUcontrol = 00 for ADD, 01 for AND, 10 for SUB, 11 for OR
  Delay = 4
- **EXTENDER (Sign/Zero) => Sign extender**
  Delay = 1
- **MUX (Select) => 2 input mux**
  Delay = 1
- **MEMORY (WrEnable, Addr) => Ideal memory**
  Delay = 1
- **REGISTER (Enable) => Clocked register**
  Clk-to-Q Delay = 1
- **REGISTER FILE (RD, RS, RT, WrEnable) => Register file**
  Read delay = 1, Setup time = 1, Hold time = 1
- **CONSTANTS => A 32 bit constant can be defined as an input to any block (no delay)**
- **SHIFTER (ShAmt) => 64 bit shifter (see symbol below)**
  – Input and output is through 2 buses which connect to the upper and lower 32 bits of a 64 bit word.
  Delay = 1



a) Draw the datapath showing all interconnections and components (including the controller).



b) What is the critical path ?

The critical path is stressed on the ADD and ADDIU instructions. It includes the PC, instruction memory, register file, the ALUSrc mux, the ALU, the WrSrc mux, and the setup time for writing to the register file.

c) What is the delay of the critical path?

The sum of all the delays above is 10. Don't forget the clock-to-Q of the PC and the setup time of the register file!

d) Show the values of all the control points for each instruction. (The Enable for the PC is given as an example)

	PCEnable	ALUSrc	ALUOp	Sign/Zero	Rotate	WrSrc
ADDIU	1	0	00	Sign	X	0
ADD	1	1	00	X	X	0
SRL	1	X	XX	X	0	1
Rotate	1	X	XX	X	1	1

Problem 4: Multi-cycle Processors (25 points)

For this problem you will be working with the multi-cycle datapath components on the next page. All inputs for the functional units are labeled, and registers only have one data input and one data output (you should not draw the clock lines). You will not need to deal with control in this problem, so the control inputs to each block are not shown.

a) Given the datapath components on the next page, determine the register transfer language description for each of the standard MIPS instructions in the table below. You do not need to fill in every row. *Hint: Do not write to the register file at the end of a cycle (i.e. – only write directly from a register, not a functional unit).*

ADDU	LW	SW	JAL
IR ← Mem(PC); PC ← PC + 4;	IR ← Mem(PC); PC ← PC + 4;	IR ← Mem(PC); PC ← PC + 4;	IR ← Mem(PC); S ← PC + 4;
A ← Reg(IR[rs]); B ← Reg(IR[rt]);	A ← Reg(IR[rs]);	A ← Reg(IR[rs]); B ← Reg(IR[rt]);	Reg(31) ← S; PC ← PC[31:26] || IR[j] || 00;
S ← A + B;	S ← A + Ext(imm);	S ← A + Ext(imm);	
Reg(IR[rd]) ← S;	M ← Mem(S);	Mem(S) ← B;	
	Reg(IR[rt]) ← M;		

b) You'll notice that some components need to be reused during execution of an instruction. Wire the datapath to support all four instructions, adding only muxes as needed. You may provide constants as inputs to any component. Be sure to label special buses, such as instruction fields. You do not need to draw any control signals (including mux select signals) – just assume they will be correctly generated in all cases.

c) For each instruction in part (a), calculate the CPI and indicate on the table above which operations occur during each cycle.

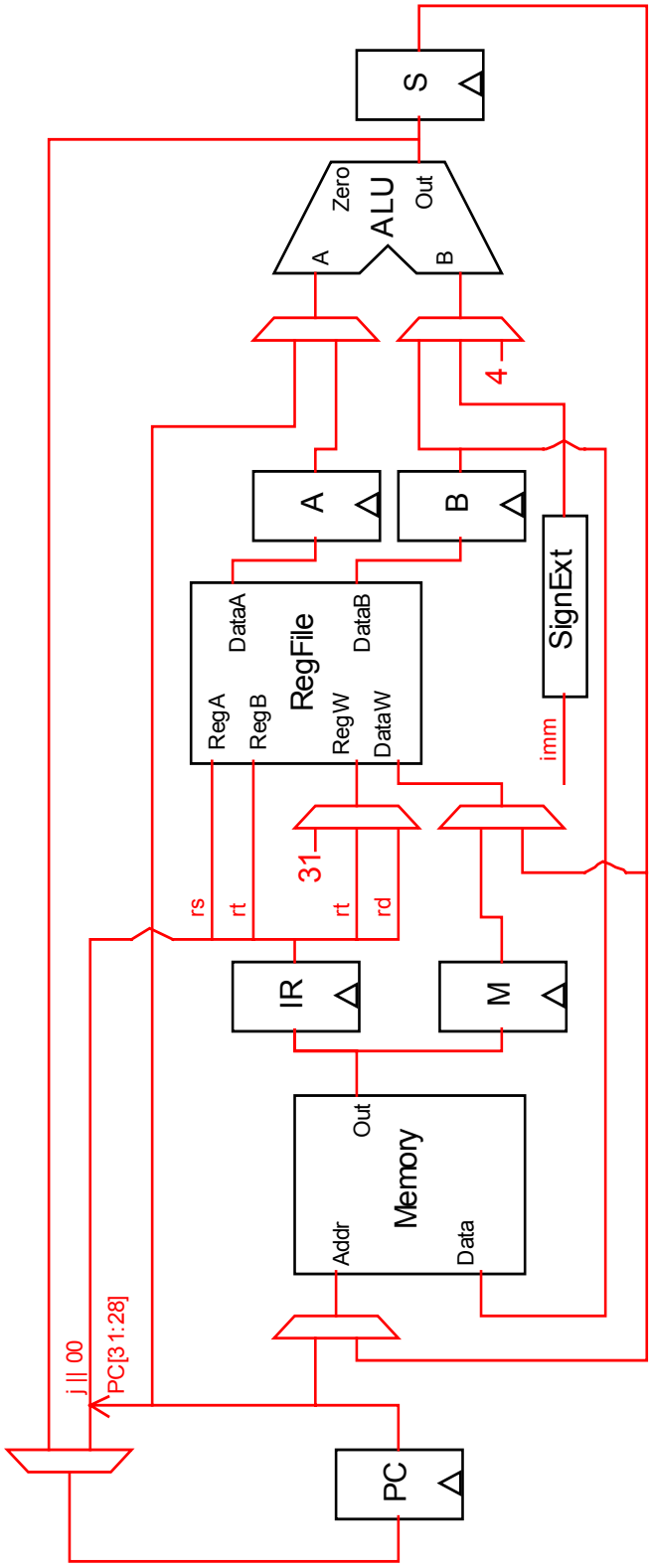
The register transfers in part (a) above are already separated into the operations that can be performed in each clock cycle. This corresponds to the following CPI: ADDU – 4, LW – 5, SW – 4, and JAL – 2.

d) The table below indicates the worst case delay through each of the functional units used in the datapath. Given these delays, calculate the execution time of this processor for a program consisting of 400,000 adds, 250,000 loads, 250,000 stores, and 100,000 branches.

Functional Unit	Worst-case Delay
Memory	50ns
Register File (read)	25ns
Register File (write)	15ns
ALU	30ns
All others	0ns

The trick to this question was realizing that cycle time is only affected by the longest path between registers in the datapath. In this case, since there is no setup time or clock-to-Q delay, the cycle time will be 50ns. There is no need to have two functional units in series in this datapath, and doing so will only reduce performance.

The execution time of the processor will be the total number of cycles required multiplied by the cycle time:
$Time = (400{,}000 \times 4 + 250{,}000 \times 5 + 250{,}000 \times 4 + 100{,}000 \times 2) \times 50ns = 202.5ms$


S
ALU
Zero
Out
A
B
4
A
B
SignExt
RegFile
DataA
DataB
RegA
RegB
RegW
DataW
imm
31
rs
rt
rt
rd
IR
M
Memory
Out
Addr
Data
j || 00
PC[31:28]
PC