CS 3725 Jan. 24/12

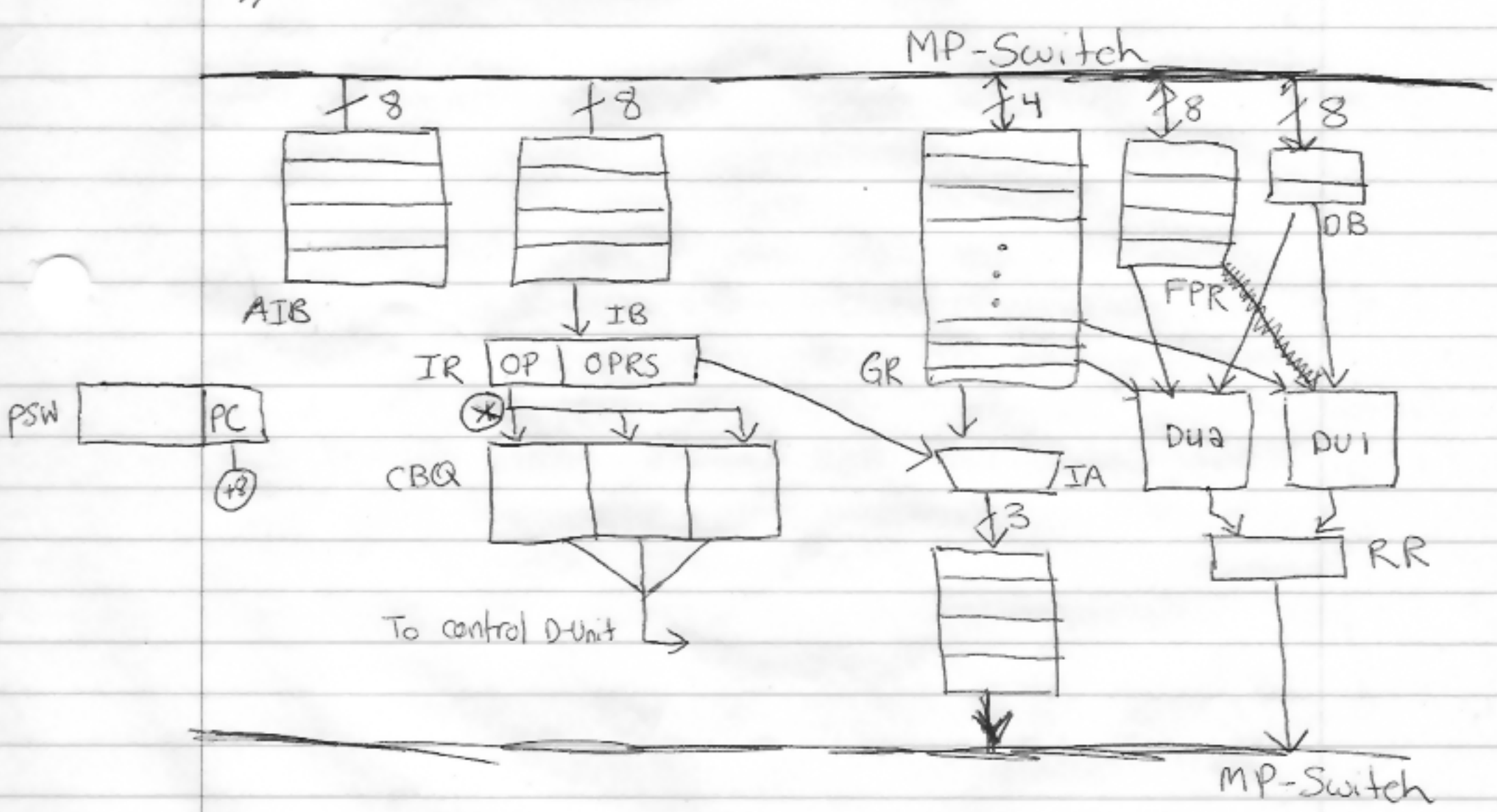
Look ahead processors

I-U
- Prefetching instructions
- Advance op code decoding
- Advance operand address calculation
- Advance operand fetch

D-U
- Multiple execution units

Instruction Pre-Processing or Inst. Preparation or Inst. Issue Stage

Ex// Look ahead processor - IBM 370/165



MP-Switch: Bus between memory and processor

- IB: (Instruction Buffer) 4 Registers, each register holds 1 instruction, instruction size is 4 bytes, memory physical word size is 8 bytes, each memory word holds 2 instructions, FIFO buffer

- IR: (Instruction Register) the bottom most instruction from IB enters IR, IR size is 4 bytes

- ⊛: Actually generates the address location in the n-program memory, relevant to the op-code in question

- CBQ: (Circular Buffer Queue) Holds 3 "decoded" op-code
  ↳ address of n-prog
  CBQ dispatches the control signals for an instruction as and when needed.
  ↳ meaning an instruction to the D-Unit if it can be executed, otherwise the opcode for that instruction is circulated in CBQ until it can be dispatched

- ARB: (Effective operand address buffer) consists of 4 registers, they are used to pre-fetch the operands (operands are pre-feteched in DB, FPR and GR) each register in ARB = 3 bytes

- DB: (Data Buffers) used for operand preload, 2 registers, reg size = 8 bytes, can store 2 operands / registers

- FPR: (Floating Point Registers) 4 of 8 byte registers, Hold floating point operands

GR: (General Purpose Registers) 16 of 4 bytes, both FPR and GR can be used directly by D-Units, GR is also used by IA to calculate (BASE + OFFSET)

IBM 370/165 uses
- Overlap
- Parallel hardware

*NOTE: -1 D-Unit is used for fast instructions (eg. add, sub, shift, etc)
- Other is used for slow instructions (eg. mult, div, etc)

- AIB (Auxiliary Instruction Buffer)
  Precluse Problem with branch instructions in a pipeline

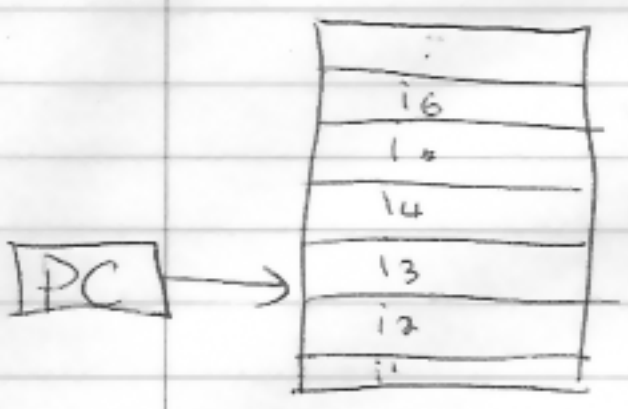
AIB and IB holds the instruction in the true branch and instruction in the false branch respectively. If its current branch instruction succeeds (ie we do take branch) then its contents of AIB and IB are swapped and then the "correct" instruction should proceed through the pipeline

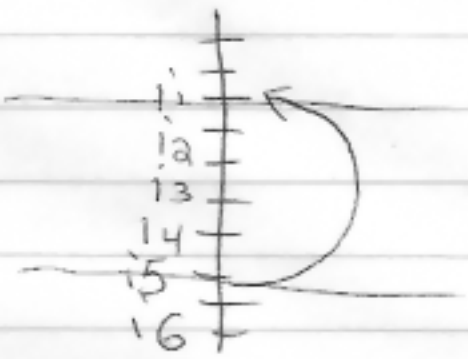
Minimizing the use of MP-Switch

- Cache - which exploits the locality of reference

- Look-ahead-look-behind registers used to "efficiently" execute loops

look ahead look behind buffer (cache)





Unlike IB in the LALB, buffers the instructions are maintained in, its only the PC, is