Exceptions and Interrupts

- Control of machine has been via PC so far

- Exceptions or Interrupts are unexpected changes in control

- Examples:
  - I/O device service request (external)
  - OS call from program (internal)
  - arithmetic exception (internal)
  - undefined instruction (internal)
  - hardware error (either)
• we consider arithmetic overflow and undefined instruction

• must generate a signal that can be used as input (along with the op code) to the control FSM

• signal will be used with current state to determine successor state that will handle exception

• therefore, an exception is detected during a particular state and the signal is available combinatorially at the end of the cycle to be used as FSM input at the next clock edge
• ALU has a condition signal that indicates overflow (like the zero condition signal used for BEQ)

• it is available combinatorially at the end of the cycle that performs state 6

• an invalid op code can be detected via combinational logic during the cycle performing state 1.

• therefore, the successor states of states 1 and 6 could be the exception handlers (the text respects only one of these)
Action to service Exception

- Save PC of instruction generating the exception (requires computation of PC - 4) – this gets much more complicated in later chapters

- Save the cause of the exception

- Set PC to exception handling routine – it may be specific to the exception or general
Extra Hardware

- Exception PC (EPC) register – fed by ALU-Out register
- CAUSE register
- CAUSE register MUX
- wider PC MUX
- EPCWrite, CauseWrite write control signals
- IntCause exception cause selection
Note on HW and FSM figures

- overflow line is not shown as input to control FSM as it should be

- invalid op signal used to detect successor state to state 1 which is as early as possible

- overflow signal is not used until after the write of the registers – state is effected by erroneous execution, this is a problem

- HW diagram does not show how overflow signal is kept until or accessed during cycle performing state 7

- overflow should be used to determine successor state of state 6.