I Flare CPU Instruction Cache Interface

I.A `ibus` Interface

```
valid:    out Bool()
ready:    in Bool()
addr:     out UInt(32 bits)
devData:  in UInt(16 bits)
```

- This is using `valid`/`ready` handshaking, but `valid` does not have to be driven in a registered fashion (so unlike AXI I guess?).
- Pseudocode fragments:
  - ```
    when (ID.up.isValid) {
      when (
        ID.rMultiCycleState === MultiCycleState.PRIMARY;
        and we have a group 7 instruction that has not had an `ID.up.isFiring`
        yet;
      ) {
        psIdHaltIt := True;
        ID.haltIt();
        ID.rMultiCycleState := MultiCycleState.G7_SUB_DECODE;
        ...
      }
    }
    ```
  - ```
    when (io.ibus.ready) {
      do C-style `union`-like instruction decoding;
      ID.dontHaltIt();
    } otherwise {
      ID.haltIt();
    }
    ```
ibus.valid	ibus.ready	psIdHaltIt	#	IF.haltIt	ID.haltIt
0	0	0	#		