Xicu component in Flattened device trees

?Xicu is a m->n irq router.

It concentrates 3 types of interrupt sources:

- IPIs (WTI)
- Timers (PTI)
- External IRQ lines (HWI)

These inputs can be filtered independently for each output signal.

MutekH context

In MutekH, we preferably use the Xicu with:

- N IPIs: 1 per processor
- 1 timer
- M Input IRQ lines (as needed)
- N output lines: 1 per processor

MutekH device tree representation

In MutekH device tree, Xicu is quite tricky to reprensent; we can enumerate the following logical devices:

- a (multi) timer
- 1 ICU (with input selection, masking and callbacks) per CPU.

Thus we use the following device tree:

- Xicu root (a timer)
  - Xicu filter for output 0
  - Xicu filter for output 1
  - ?

Flattened device tree representation

In FlattenedDeviceTree, the representation follows this logic.

- There is a timer device which is the root
- There is one filter per output line.

```
xicu@0xd2200000 {
    device_type = "soclib:xicu:root";
    input_lines = <2>;
    ipis = <4>;
    timers = <1>;
    reg = <0xd2200000 0x1000>;

    out@0 {
        device_type = "soclib:xicu:filter";
        parent = &{/xicu@0xd2200000};
        output_line = <0>;
```

```
            irq = <&{/cpus/Mips,32@0} 0>;
        };

        out@1 {
            device_type = "soclib:xicu:filter";
            parent = &{/xicu@0xd2200000};
            output_line = <1>;
            irq = <&{/cpus/Mips,32@1} 0>;
        };

        ?
    };
```

Timer designation

Timer may be designated by a couple `<&{device_path} timer_id>`, maximum timer id is defined by the xicu root `"timers"` property.

Timer may be used in `/chosen`, like:

```
    chosen {
        console = &{/tty@0xd0200000};
        timer = <&{/xicu@0xd2200000} 0>;
    };
```

Hardware IRQ input routing

As usual, other components route their IRQs to targets; but as the *root* is a timer device, IRQs must be routed to one of the output filters. Couple is: `<&{device_path} input_id>`, maximum hardware line id is defined by the xicu root `"input_lines"` property.

```
    tty@0xd0200000 {
        device_type = "soclib:tty";
        tty_count = <1>;
        reg = <0xd0200000 0x10>;
        irq = <&{/xicu@0xd2200000/out@0} 0>;
    };
```

IPI routing

Likewise, in MutekH, IPIs are handled by the ICU device class. Processors must use xicu filters as IPI controllers. Couple is: `<&{device_path} ipi_id>`, maximum IPI id is defined by the xicu root `"ipis"` property.

```
    Mips,32@0 {
        name = "Mips,32";
        device_type = "cpu";
        reg = <0>;
        icudev_type = "cpu:mips";
        ipi = <&{/xicu@0/out@0} 0>;
    };
```