Pulser Circuit

Marie Farrell

Department of Computer Science
Maynooth University

Here, we present a modular $\mathcal{EVT}$-specification corresponding to the Event-B specification of a single pulser circuit as presented in Chapter 8 of Abrial's book.

```
spec  DATAM0 =
  ops circuit : Bool
      push, pop, flash : ℕ
  . pop ≤ push
    push ≤ pop + 1
    flash ≤ push
    push ≤ flash + 1
    circuit = FALSE
        ⇒ flash = push ∨ flash = pop
end

spec  ENV0 =
  DATAM0 then
  Events
    env =
      when circuit = FALSE
      thenAct circuit = TRUE
end

spec  CIR0 =
  ENV0 with {env ↦ cir, FALSE ↦ TRUE,
              TRUE ↦ FALSE}
end
```

```
spec  M0 =
  ENV0 with {env ↦ env2}
  and ENV0 with {env ↦ env3}
  and CIR0 with {cir ↦ cir1}
  and CIR0 with {cir ↦ cir2}
  then
    Events
      Init =
        thenAct circuit := FALSE
                push := 0
                pop := 0
                flash := 0
      env1 =
        when pop = push
        thenAct push := push + 1
     env2 =
       when pop ≠ push
       thenAct pop := pop + 1
     cir1 =
       when push ≠ flash
       thenAct flash := flash + 1
     cir2 =
       when push ≠ pop ∨ push = flash
end
```

Fig. 1. The abstract model

```
spec  M_1_1 =
  M0 then
  . circuit = FALSE ∧ pop ≠ push
      ⇒ flash ≠ pop
    circuit = FALSE ∧ pop ≠ push
      ⇒ flash = push
  Events
    cir2 =
     when push = flash
end
```

```
spec  M_1_2 =
  M0 then
  . pop ≠ push ⇒ flash ≠ push
    pop ≠ push ⇒ flash = pop
  Events
    cir1 =
      when push = pop
end
```

Fig. 2. The first refinement

```
spec DATAM2 =
  ops input, output, reg : Bool
  . input = TRUE ⟺ pop ≠ push
    circuit = FALSE ⇒ reg = input
  Events
    Init =
      thenAct input := FALSE
              output := FALSE
              reg := FALSE
end

spec ENV21 =
  DATAM2 then
  Events
    env =
      when input = FALSE
      thenAct input := TRUE
end

spec CIR21 =
  DATAM2 then
  Events
    cir =
      thenAct output := TRUE
              reg := input
end
```

```
spec M_2_1 =
  M_1_1 and DATAM2 and
  ENV21 with {env ↦ env1} and
  ENV21 with {env ↦ env2, FALSE ↦ TRUE,
              TRUE ↦ FALSE} and
  CIR21 with {cir ↦ cir1} and
  CIR21 with {cir ↦ cir2, TRUE ↦ FALSE}
  then
    . circuit = TRUE
        ⇒ (input = TRUE ∧ reg = FALSE ⟺
push ≠ flash)
    Events
      cir1 =
        when input = TRUE
             reg = FALSE
      cir2 =
        when input = FALSE ∨ reg = TRUE
end

spec M_2_2 =
  M_1_2 and DATAM2 and
  ENV21 with {env ↦ env1} and
  ENV21 with {env ↦ env2, FALSE ↦ TRUE,
              TRUE ↦ FALSE} and
  CIR21 with {cir ↦ cir1} and
  CIR21 with {cir ↦ cir2, TRUE ↦ FALSE}
  then
    Events
      cir1 =
        when input = FALSE
             reg = TRUE
      cir2 =
        when input = TRUE ∨ reg = FALSE
end
```

Fig. 3. The second refinement step

```
spec REFCIR1 =
  DATAM2 then
  Events
    cir =
      thenAct output := bool(input = TRUE
                        ∧ reg = FALSE)
              reg := bool(input = TRUE)
end

spec REFCIR2 =
  DATAM2 then
  Events
    cir =
      thenAct output := bool(input = FALSE
                        ∧ reg = TRUE)
              reg := bool(input = TRUE)
end
```

```
spec M_3_1 =
  M_2_1 and
  REFCIR1 with {cir ↦ cir1} and
  REFCIR1 with {cir ↦ cir2}
end

spec M_3_2 =
  M_2_2 and
  REFCIR2 with {cir ↦ cir1} and
  REFCIR2 with {cir ↦ cir2}
end
```

Fig. 4. The third refinement step

```
spec  M_4_1 =
  M_3_1 with {cir1 ↦ pul1, cir2 ↦ pul1}
end

spec  M_4_2 =
  M_3_2 with {cir1 ↦ pul1, cir2 ↦ pul1}
end
```

Fig. 5. The fourth refinement step