

[HW2_Report] FIR + 8051 Co-design

設計想法

1-Cycle

N-Cycle

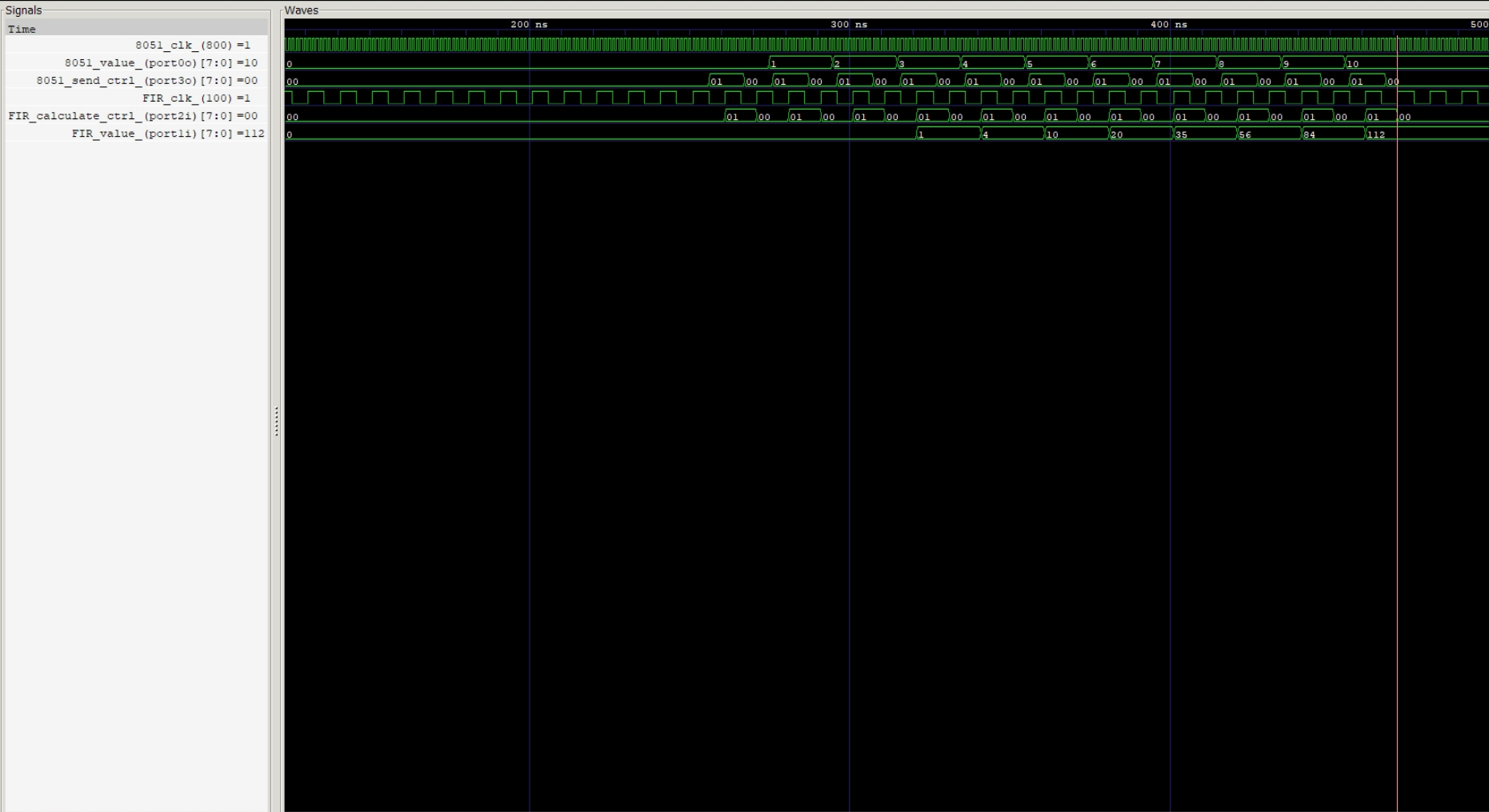
1-Cycle 和 N-Cycle 比較

設計想法

一開始也是照助教給的 8051 code 控制 8051 ISS，但 FIR 和 8051 的 clock 不一樣，所以會出現訊號對不上的問題，因此將 P2 改成計算完成的訊號，讓 8051 等待 FIR 計算完成後才將 P3 控制訊號關閉，再重新進入下一個計算循環因此，新的 FIR module I/O 配置條列如下：

- A：P0, 8051 → FIR, Prepared value which needs calculate. (port0o)
- B：P3, 8051 → FIR, Tell FIR can catch value. (port3o)
- C：P2, FIR → 8051, FIR calculate completed or not. (port2i)
- D：P1, FIR → 8051, FIR send result value back. (port1i)

1-Cycle



N-Cycle



1-Cycle 和 N-Cycle 比較

```
/* 1-Cycle */

sum = c[0] * A.read() +
      c[1] * shift[0] +
      c[2] * shift[1] +
      c[3] * shift[2] +
      c[4] * shift[3] +
      c[5] * shift[4] +
      c[6] * shift[5] +
      c[7] * shift[6] +
      c[8] * shift[7];

D.write(sum);
```

```
/* N-Cycle */

sum += c[0] * A.read();
wait();
sum += c[1] * shift[0];
wait();
sum += c[2] * shift[1];
wait();
sum += c[3] * shift[2];
wait();
sum += c[4] * shift[3];
wait();
sum += c[5] * shift[4];
wait();
sum += c[6] * shift[5];
wait();
sum += c[7] * shift[6];
wait();
sum += c[8] * shift[7];

D.write(sum);
sum = 0;
```

右邊程式碼使用 wait() 將 FIR 拆成數次運算以實現 N-Cycle 的功能（即每次只能計算一組係數和資料的乘積再累加至 sum 中），相較於 1-Cycle，由於每次只進行一組運算，因此在硬體上可視為只使用一個加法器和一個乘法器，可節省硬體空間，但須花費更多時間運算，而 1-Cycle 因為一次使用多個乘法器和加法器可在一個周期內完成運算，運算速度快，但較花費硬體空間及成本。從兩者的波形圖可觀察兩者的速度差異，在運行相同 8051 ISS 程式碼的條件下，1-Cycle 大概只花 200ns 就將整個運算做完，但 N-Cycle 卻需要花費 1100 ns 才可完成。輸出結果稍有不同是因為在執行一個完整的 FIR 運算後有將 sum 歸零，有用手算過數值是沒問題的。