**第一行SW指令的執行：**

我們首先分析第一行sw指令的執行情況，該指令將值0x0000FFFF寫入GPIO啟用暫存器。

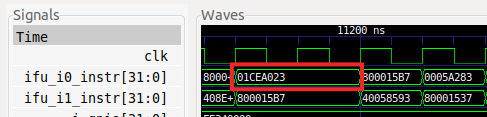
* 組合語言指令： **sw t3,0(t4)**
* 機器指令： **0x01cea023**

將時間範圍設定為11100 ns-11800 ns 

在約11200 ns處取儲存指令（**0x01cea023**），如訊號*ifu\_i0\_instr*所示（請參閱圖1）。訊號字首表示它是指令擷取單元（*ifu*）的一部分。該訊號位於2向超標量處理器的通路0（\_*i0*），並且是待取指令（*\_instr*）。

幾個週期（在此期間，在CPU中對指令進行解碼並執行指令…）後，將寫入請求傳送到I/O系統，如圖1所示。具體來說，在約11500 ns處：

* CPU通過Wishbone匯流排傳送要寫入的位址（*wb\_m2s\_io\_adr*=0x00001408）。使用訊號*wb\_io\_adr\_i*=0x00001408將該位址提供給多工器。
* 多工器基於位址0x00001408選擇GPIO從屬裝置（*match = 0000010*且   
  *wb\_gpio\_cyc\_0*=1），從而將其所有訊號連接到與CPU相連的Wishbone匯流排。具體來說：
  + **wb\_gpio\_dat\_o = wb\_io\_dat\_i = 0x0000FFFF**（通過儲存指令提供給GPIO的值）
  + **wb\_gpio\_adr\_o = wb\_io\_adr\_i = 0x00001408**（提供給GPIO的位址，與啟用暫存器相對應）。
  + 最後，在多工器做出選擇後1個週期，啟用暫存器（*ext\_padoe\_o*）用儲存指令提供的值更新：



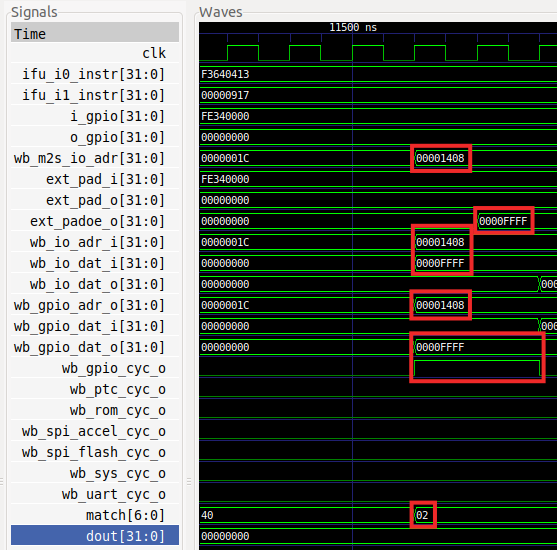


圖1. 模擬啟用暫存器寫入操作

**LW指令的執行：**

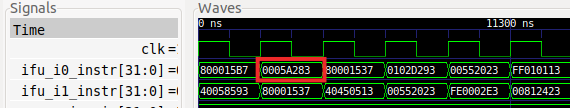
我們現在分析讀取開關值的lw指令的執行情況。

* 組合語言指令： **lw t0,0(a1)**
* 機器指令： **0x0005a283**

同樣，將時間範圍設定為11100 ns-12000 ns。在約11200 ns處取載入指令（**0x0005a283**），如*ifu\_i0\_instr*所示（請參閱圖2）。

幾個週期（在此期間，在CPU中對指令進行解碼並執行指令）後，將讀取請求傳送到I/O系統。具體來說：

* 開關值通過訊號*i\_gpio*和*ext\_pad\_i*（可從圖2中查看程式碼）提供給GPIO模組。可以在圖中驗證測試平台中模擬的開關值是0xFE34，該值是訊號*i\_gpio[31:16]*和*ext\_pad\_i[31:16]*中包含的值。
* CPU通過Wishbone匯流排傳送要寫入的位址（*wb\_m2s\_io\_adr*=0x80001400）。使用訊號*wb\_io\_adr\_i*=0x80001400將該位址提供給多工器。
* 多工器基於位址0x00001400選擇GPIO從屬裝置（*match = 0000010*且   
  *wb\_gpio\_cyc\_0*=1），從而將其所有訊號連接到與CPU相連的Wishbone匯流排。特別是在模擬中，可以看到：
  + **wb\_io\_dat\_o = wb\_gpio\_dat\_i = 0xFE340000**（GPIO由載入指令提供  
    的值）。
  + **wb\_gpio\_adr\_o = wb\_io\_adr\_i = 0x00001400**（提供給GPIO的位址，與讀暫存器相對應）。
* 最後請注意，幾個週期後，暫存器t0（模擬中的訊號*dout*）使用從開關讀取的值更新：16



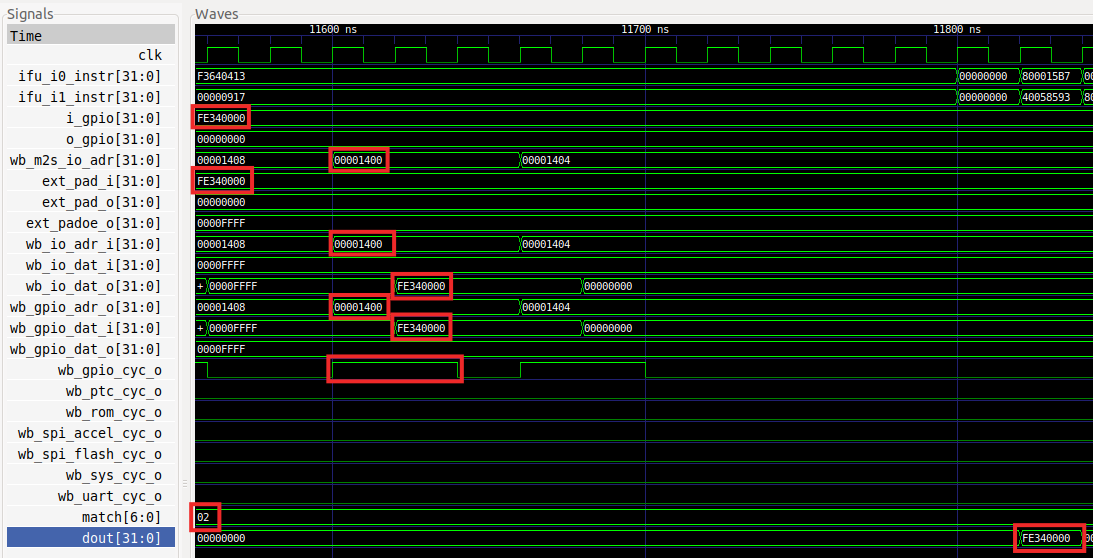


圖2. 開關讀取模擬

**第二條SW指令的執行：**

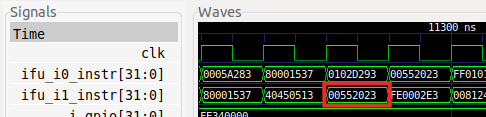
最後，我們分析第二條sw指令的執行情況，該指令設定LED值。

* 組合語言指令： **sw t0,0(a0)**
* 機器指令： **0x00552023**

將時間範圍設定為11200 ns-12300 ns。在約11300 ns處，在*ifu\_i1\_instr*中擷取儲存指令（**0x00552023**）（請參閱圖3)。

幾個週期（在此期間，在CPU中對指令進行解碼並執行指令）後，將寫入請求傳送到I/O系統。分析圖3遵循以下步驟：

* CPU通過Wishbone匯流排傳送要寫入的位址（*wb\_m2s\_io\_adr*=0x80001404）。使用訊號*wb\_io\_adr\_i*=0x80001404將該位址提供給多工器。
* 多工器基於位址0x00001404選擇GPIO從屬裝置（*match = 0000010*且   
  *wb\_gpio\_cyc\_0*=1），從而將其所有訊號連接到與CPU相連的Wishbone匯流排。具體來說：
  + **wb\_gpio\_dat\_o = wb\_io\_dat\_i = 0x0000FE34**（通過儲存指令提供給GPIO的值）
  + **wb\_gpio\_adr\_o = wb\_io\_adr\_i = 0x00001404**（提供給GPIO的位址，與啟用暫存器相對應）。
  + 最後，在多工器做出選擇後1個週期，*ext\_pad\_o*用儲存指令提供的值更新：**ext\_padoe\_o=0x0000FE34**。該值通過訊號**o\_gpio=0x0000FE34**提供給LED。



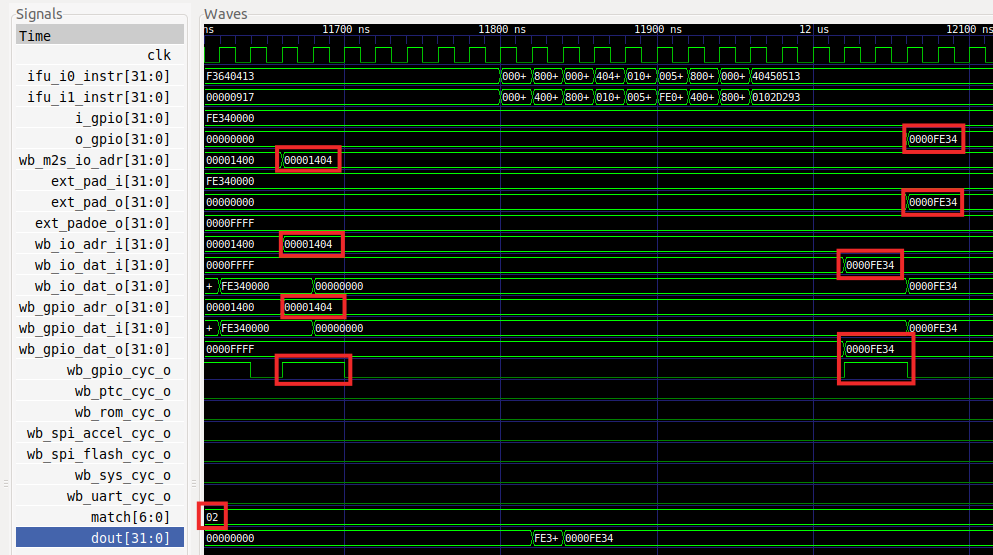


圖3. LED寫入模擬