

Si se transforma la secuencia anterior en instrucciones VLIW genéricas, se obtiene el siguiente fragmento de código:

```
LD F0, 0(R1)
LD F0, -8(R1)
LD F0, -16(R1)    ADDD F4, F0, F2
LD F0, -24(R1)   ADDD F4, F0, F2
LD F0, -32(R1)   ADDD F4, F0, F2
SD 0(R1), F4     ADDD F4, F0, F2    LD F0, -40(R1)
SD -8(R1), F4    ADDD F4, F0, F2
SD -16(R1), F4   ADDD F4, F0, F2
SD -24(R1), F4
SD -32(R1), F4
SD -40(R1), F4
```

Observe que si el formato de la instrucción VLIW no se ajustase al patrón, por ejemplo, por disponer solo de una operación de acceso a memoria, habría problemas ya que el patrón habría que descomponerlo en dos instrucciones VLIW al contar con una operación de carga y otra de almacenamiento. Esto produciría la pérdida de datos al romperse la secuencialidad en el flujo de datos:

```
LD F0, 0(R1)
LD F0, -8(R1)
LD F0, -16(R1)   ADDD F4, F0, F2
LD F0, -24(R1)   ADDD F4, F0, F2    // Se pierde su resultado
LD F0, -32(R1)   ADDD F4, F0, F2
SD 0(R1), F4     ADDD F4, F0, F2
LD F0, -40(R1)   // Nadie lee el valor de F4
SD -8(R1), F4    ADDD F4, F0, F2    // ya que se machaca el valor previo de F4
SD -16(R1), F4   ADDD F4, F0, F2
SD -24(R1), F4
SD -32(R1), F4
SD -40(R1), F4
```