

a)

```

LD F0, 0(R1) // Carga de X[i]
ADDD F4, F0, F2 // X[i] = X[i] + a
LD F0, -8(R1) // Carga de X[i-1]
SUBI R1, R1, #16
inicio: SD 16(R1), F4 // Almacenamiento de X[i]
ADDD F4, F0, F2 // X[i-1] = X[i-1] + a
LD F0, 0(R1) // Carga de X[i-2]
SUBI R1, R1, #8
BNEZ R1, inicio // ¿Fin del bucle: R1=0?
SD 16(R1), F4 // Almacenamiento de X[2]
ADDD F4, F0, F2 // X[1] = X[1] + a
SD 8(R1), F4 // Almacenamiento de X[1]

```

b) El código VLIW sería el siguiente:

	Memoria	Coma flotante	Entero/salto
prólogo:	LD F0, 0(R1)	-	-
	LD F0, -8(R1)	-	-
	-	ADDD F4, F0, F2	SUBI R1, R1, #16
	-	-	-
inicio:	SD 16(R1), F4	ADDD F4, F0, F2	SUBI R1, R1, #8
	LD F0, 8(R1)	-	BNEZ R1, inicio
	-	-	-
epílogo:	SD 16(R1), F4	ADDD F4, F0, F2	-
	-	-	-
	-	-	-
	SD 8(R1), F4	-	-

El total de ciclos ejecutados para procesar un vector de 1000 elementos sería de 3009 ciclos (5+1000*3+ 4).

c) No se aprovechan correctamente los cauces de las unidades funcionales ya que en un momento dado solo hay una operación en cada unidad funcional: solo se admite un dato en las unidades funcionales, no tantos como segmentos tienen. Ello es debido a que si no se consideran las latencias de las unidades, la técnica no se aplica de forma óptima. El patrón correcto se obtiene reflejando las latencias de las unidades:

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

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
```