

EA075 – Primeira Avaliação

Profa. Ting

23/09/2019

RA: _____ Nome: _____

Questão:	1	2	3	4	Total
Pontos:	7	9	9	10	35
Resultado:					

A nota final seria a soma dos pontos dividida por 2,7.

Dado um sistema computacional com as seguintes características:

1. um microprocessador de endereçamento de 24 *bits* e registradores gerais e aritméticos de 16 *bits*,
2. um sistema de memória constituído por 128K de ROM, 64K de RAM estática e 1M de DRAM, onde a ordenação é *little endian*,
3. um disco rígido magnético SATA, e
4. dois periféricos mapeados no espaço da memória, um conversor AD e um conversor DA

Sob o ponto de vista da linguagem C, o processador suporta tipos de dados de 8 *bits* (char), 16 *bits* (int) e 32 *bits* (long int). Com exceção do tipo de dados char, a representação dos dados é com sinal, em complemento para dois. Quando se declara uma variável com o qualificador unsigned, o dado é representado internamente sem o *bit* de sinal.

1. Microprocessadores:

- (a) (1pt) Considere que o processador tenha 7 registradores e uma instrução com 3 campos: um campo de 5 *bits* para o código de operação, um campo para o registrador onde é armazenado um dos operandos, e um campo para o valor (imediato) sem sinal do segundo operando. Qual é o valor imediato representável? Justifique.

- (b) (2pt) Foram alocados para as variáveis a, b, c e d, respectivamente, os endereços 0x200000, 0x200008, 0x200010, 0x200018 ao gerar o executável do programa que contém o trecho de código abaixo. Qual é o tamanho de memória, em *bytes*, alocado para cada variável e o conteúdo de cada espaço alocado? Justifique preenchendo na seguinte tabela o conteúdo desses espaços de memória em hexadecimal, do menor endereço para o maior.

```

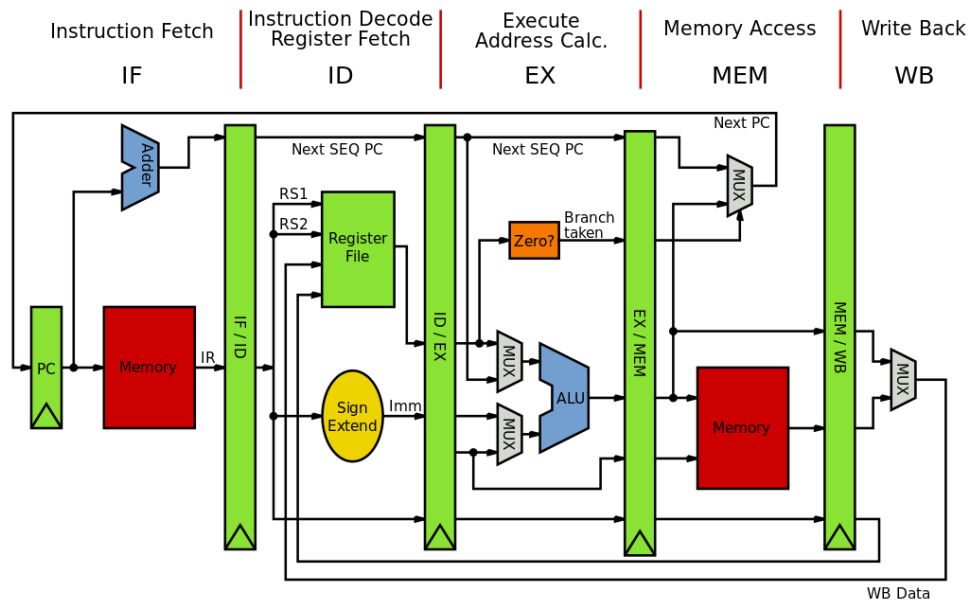
char a;      int b;      long int c;   char *d;
a = 31;     b = -15;     c = 2048;   d = &a;

```

Var	Tamanho (bytes)	Conteúdo dos endereços	Justificativa
a			
b			
c			
d			

(c) (1pt) Para programar o processador, é necessário instalar um sistema operacional? Justifique.

(d) (2pt) Instruções são segmentadas numa arquitetura *pipeline*. Elas são executadas em diferentes fases. Na arquitetura DLX, cada instrução é executada em 5 fases conforme ilustra o seguinte diagrama de blocos.



Dada uma sequência de instruções em *assembly*:

```

LW R1, 4(R31)      ;R1 = Mem[R31+4]
LW R2, 6(R31)      ;R2 = Mem[R31+6]
ADDI R3,R1,5       ;R3 = R1+5
LW R4, 0(R3)       ;R4 = Mem[R3+0]
SUB R4,R1,R4       ;R4 = R1 - R4

```

Qual é, em ciclos de relógio, o tempo médio de execução de uma instrução se o nosso processador for da arquitetura DLX e se cada fase ser executada

em um ciclo de relógio? Justifique completando a seguinte tabela. Observe que as 5 fases da primeira instrução já foram inseridas.

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
LDW R1, 4(R31)	IF	ID	EX	ME	WB															
LDW R2, 8(R31)																				
ADDI R3, R1, 5																				
LDW R4, 0(R3)																				
SUB R4, R1, R4																				

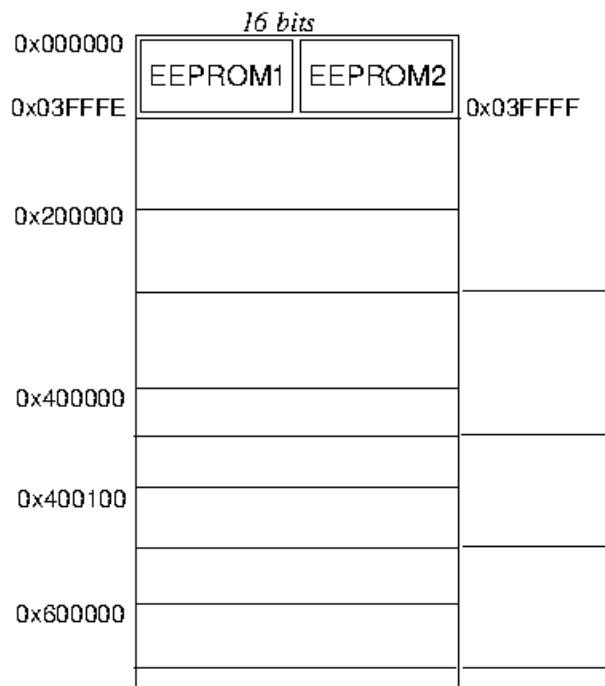
(e) (1pt) Distinguem-se três tipos de *hazards* na execução de uma sequência de instruções na arquitetura de *pipeline* DLX: estruturais, de dados e de controle. Quais tipos de *hazard* ocorrem na execução do trecho de código do item (d)? Justifique.

2. Tecnologia de memória: A memória principal é constituída pelos seguintes módulos de memória:

1. a partir do endereço 0x000000: 128K palavras de EEPROM 128Kx8.
2. a partir do endereço 0x200000: 64K palavras de SRAM 32Kx8.
3. a partir do endereço 0x400000: 64 *bytes* para registradores do periférico de E/S1.
4. a partir do endereço 0x400100: 64 *bytes* para registradores do periférico de E/S2.
5. a partir do endereço 0x600000: 1M palavras de DRAM 1Mx4.

(a) (1pt) Qual dos módulos de memória é usado para armazenar BIOS? Justifique.

- (b) (3pt) Quantos módulos de memória são necessários para cada sub-espaco da memória principal? Esboce-os na figura abaixo, preenchendo o limitante superior de cada sub-espaco em *bytes*, como foi preenchido para o subespaco correspondente ao módulo EEPROM. Justifique mostrando os cálculos que você fez para chegar às quantidades de módulos e os endereços finais de cada sub-espaco.



- (c) (1pt) Preencha a tabela de mapeamento de endereços dos módulos EEPROM, como foi feito para um módulo x de SRAM.

Dispositivo	A23	A22	A21	A20	A19	A18	A17	A16	A15	A14	A13	A12	A11	A10	A09	A08	A07	A06	A05	A04	A03	A02	A01	A00
EEPROM1																								
EEPROM2																								
SRAMx	0	0	1	0	0	0	0	1	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	1

(d) (2pt) Adotando a estratégia de decodificação completa, sintetize as funções lógicas dos sinais de ativação \overline{CS} (ativo baixo) dos módulos EEPROM. Inclua na sua função lógica \overline{AS} (ativo baixo). Este sinal indica que os endereços no barramento estão válidos.

(e) (1pt) Considere que o processador e a memória SRAM sejam, respectivamente, 68000 e CY62256. Foram implementadas as seguintes funções lógicas para os sinais de um dos módulos de CY62256 em termos dos sinais de 68000:

- $A00 - A14 \Leftarrow A01 - A15$
- $I/O0 - I/O7 \Leftarrow D08 - D15$
- $\overline{WE} = AS \vee R/\overline{W}$
- $\overline{CE} = AS \vee UDS \vee A23 \vee A22 \vee \overline{A21} \vee A20 \vee A19 \vee A18 \vee A17 \vee \overline{A16}$
- $\overline{OE} = R/\overline{W}$

Qual dos dois sinais, \overline{WE} e \overline{CE} , controla ciclos de escrita? Justifique. (Obs.: (1) O pino A00 no processador 68000 é substituído pelos pinos $\overline{LDS}/\overline{UDS}$. (2) Os diagramas de tempo do ciclo de escrita dos dois módulos estão em anexo.)

(f) (1pt) Qual dos dois HDs mostrados na figura abaixo é o usado na implementação da memória secundária? Justifique.



(a)



(b)

3. Organização de Memória: Assumindo que seja possível organizar os módulos de memória de forma que 1 módulo da SRAM (32Kx8) seja usado como a memória *cache* de mapeamento direto com linhas de *cache* de 64 bytes da memória principal projetada na questão (2). Os tempos médios de acesso às memórias *cache* e à memória principal são, respectivamente, 55ns e 110ns.

Foi programado o seguinte trecho de instruções para executar no processador:

```
int dado[5][32];
long int soma = 0;
for (char j=0; j<5; j++)
    for (char i=0; i<5; i++)
        soma += dado[i][j];
```

O vetor `dado[5][32]` foi alocado a partir do endereço `0x620000` e seus elementos são armazenados por linhas, isto é, na sequência `dado[0][0]` a `dado[0][31]`, depois `dado[1][0]` a `dado[1][31]` e sucessivamente até `dado[4][0]` a `dado[4][31]`.

- (a) (1pt) Quantos *bits* são necessários para endereçar as palavras num *slot* da memória *cache*, ou num bloco da memória principal?

- (b) (1pt) Quantos *bits* são necessários para endereçar o *slot* da memória *cache*

em que se encontra uma palavra de um dado endereço? Justifique.

(c) (1pt) Quantos *bits* são necessários para endereçar o sub-espço da memória principal em que se encontra uma palavra (*Tag*)?

(d) (1pt) De acordo com o estado da arte da tecnologia, não é ainda economicamente viável substituir integralmente HDs pelo dispositivo SSD nos computadores pessoais. Qual é uma solução aplicada para termos um melhor compromisso entre custo e benefício? Justifique.

(e) (4pt) Considere que a memória *cache* esteja resetada antes da execução da sequência de instruções programada. Qual é o tempo médio de acesso por dado? Justifique, preenchendo na tabela abaixo o que acontece em cada passo da execução, seguindo o modelo da descrição do primeiro acesso.

Acesso	Evento	Slot	Tempo	Comentários
dado[0][0]	1 <i>miss</i>	0	110ns	Bloco 100352 (0x620000) (dado[0][0]–dado[0][31]) é carregado no <i>slot</i> 0

(f) (1pt) Dado o endereço 0x6207A0, em qual *slot* da memória *cache* o seu conteúdo é carregado?

4. Transdutores e Conversores AD/DA: No nosso sistema computacional, o conversor AD está conectado com um termistor NTC (*Negative Temperature Coefficient*), e o conversor DA com um acionador de um motor.

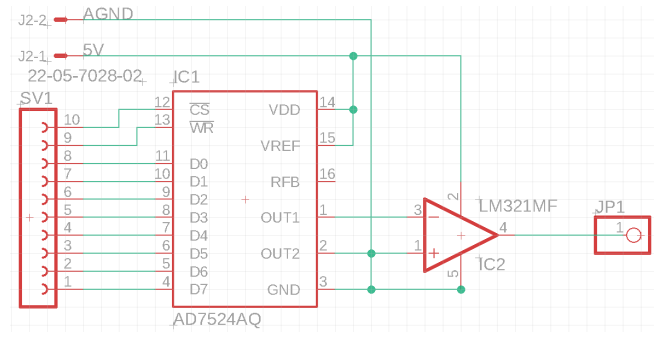
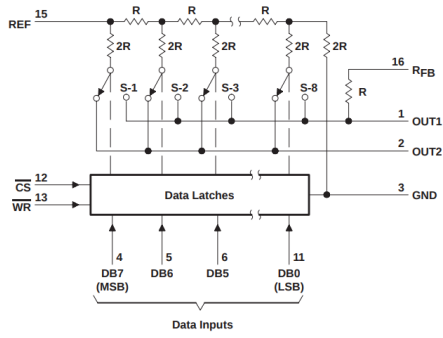
(a) (1pt) Quando se faz a conversão de um sinal analógico para um sinal digital, é preciso amostrar o sinal analógico e quantizá-lo. Cite uma técnica que assegure que os tempos de *setup* e de *hold* sejam satisfeitos na captura de uma amostra analógica que varia continuamente no tempo? Justique

mostrando o princípio desta técnica.

- (b) (1pt) Segundo a folha de dados do conversor analógico-digital ADC0801, pode-se ajustar a resolução do conversor através do pino VREF. Quando o pino VREF fica flutuando (desconectado), $V_{REF} = V_{ref}/2$ é a metade da tensão de alimentação V_{CC} , ou seja, o fundo de escala do conversor é $V_{ref} = V_{CC}$. Qual é a resolução percentual desse conversor?

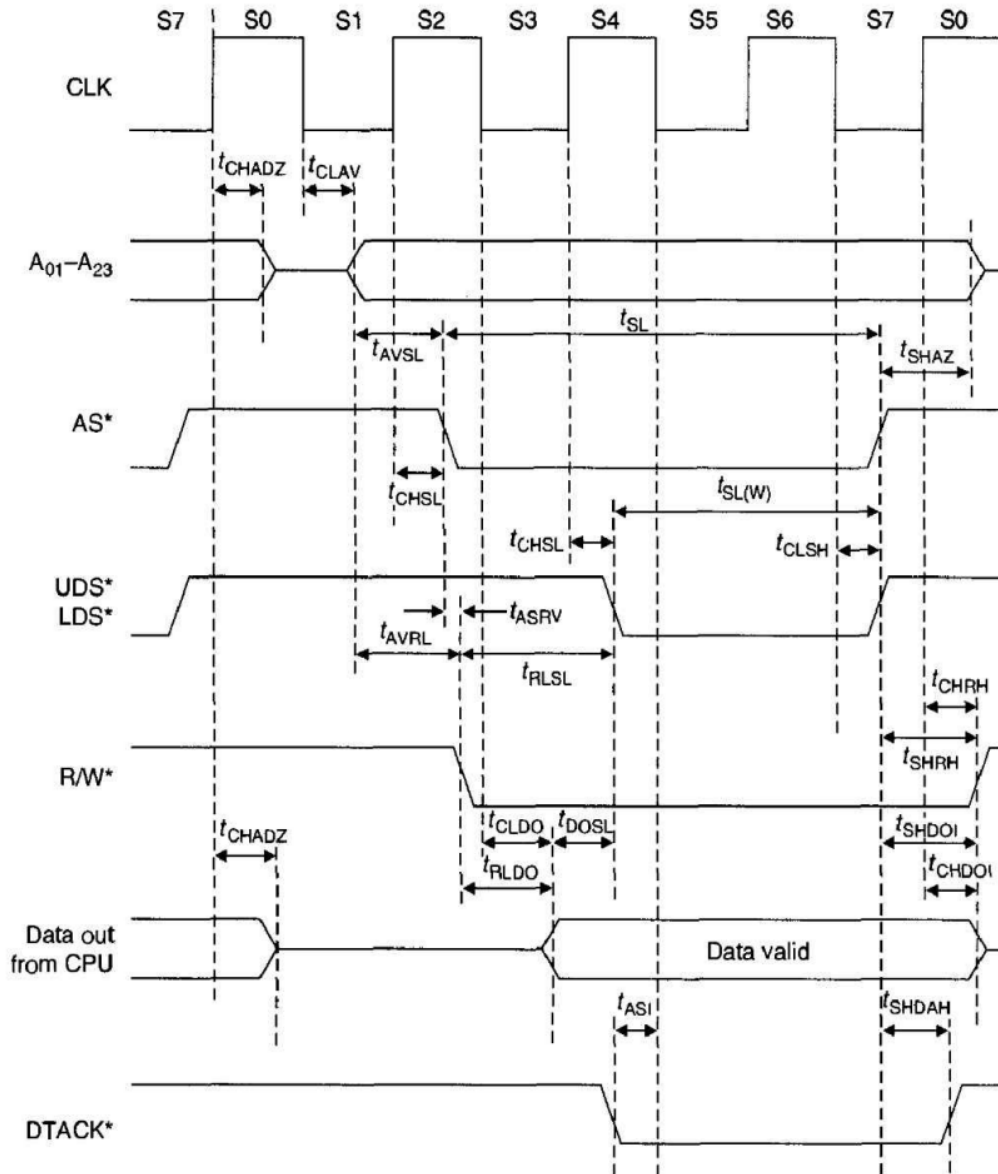
- (c) (1pt) O módulo ADC0801 é um conversor de aproximações sucessivas (SAR). Se o processamento de cada *bit* dure um ciclo de relógio, qual é o tempo de conversão, em ciclos de relógio, deste conversor?

- (d) (3pt) Um termistor é um sensor semicondutor de temperatura, capaz de gerar uma grande variação na resistência de forma proporcional a pequenas variações na temperatura. Usando um divisor de tensão, é possível medir indiretamente as resistências e, portanto, as temperaturas. Analise o seguinte esquemático e complete a conexão do termistor com a entrada analógica (pino 6) do conversor ADC0801, como parte de um divisor de tensão de 5V, e explique sucintamente como você computaria através deste circuito o valor da resistência variável do termistor.



Folhas de Dados:

1. Ciclo de Escrita do Processador 68000

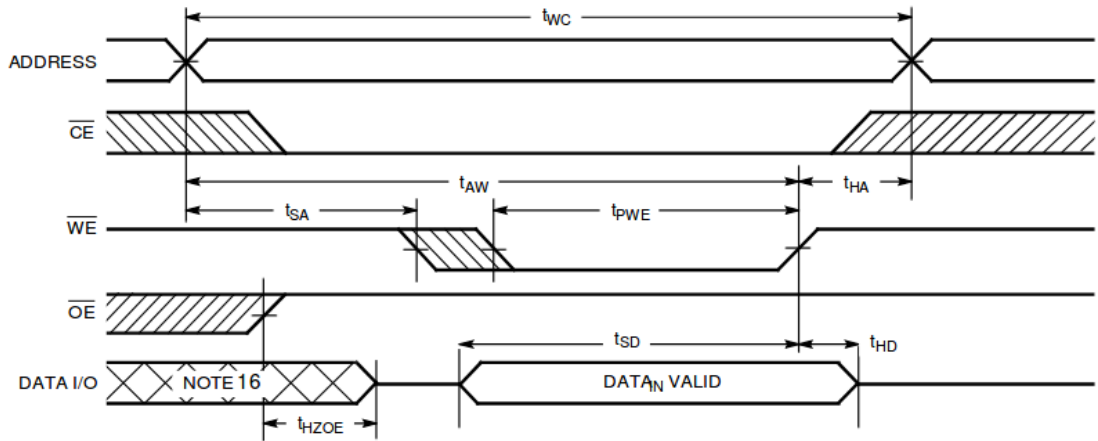


Parameter Name	Symbol	8 MHz	
		Minimum	Maximum
Clock period	t_{eye}	125	250
Clock high to data and address bus high impedance	t_{CHADZ}		80
Clock low to address valid	t_{CLAV}		62
Address valid to AS* asserted	t_{AVSL}	30	
AS* asserted	t_{SL}	270	
AS*, DS* negated to address bus high impedance	t_{SHAZ}	40	
Clock high to AS*, DS* asserted	t_{CHSL}	3	60
Clock low to AS*, DS* negated	t_{CLSH}		62
DS* asserted in write cycle	$t_{SL(W)}$	140	
AS* asserted to R/W* low	t_{ASRV}		10
Address valid to R/W* low	t_{AVRL}	20	
R/W* low to DS* asserted	t_{RLSL}	80	
Clock high to R/W* high	t_{CHRH}	0	55
AS*, DS* negated to R/W* high	t_{SHRH}	40	
Clock low to data out valid	t_{CLDO}		62
Data out valid to DS* asserted	t_{DOSL}	40	
AS*, DS* negated to data out invalid	t_{SHDOI}	40	
Data out hold from clock high	t_{CHDOI}	0	
R/W* low to data out valid	t_{RLDO}	30	
Asynchronous input setup time (DTACK* setup)	t_{ASI}	10	
AS*, DS* negated to DTACK* negated (asynchronous hold)	t_{SHDAH}	0	240

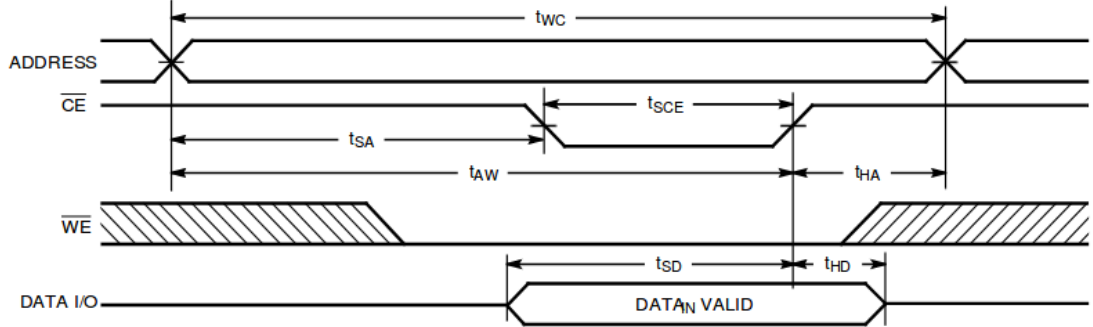
Note: Some of the parameters in this table differ from those of Table 4.4. The reader should be aware that they may change when the manufacturer publishes an updated data sheet.

2. Ciclo de Escrita da Memória CY62256 32K x 8: é controlado por \overline{WE} ou por \overline{CE} , aquele que for ativado primeiro.

Write Cycle No. 1 (\overline{WE} Controlled) [9, 14, 15]



Write Cycle No. 2 (\overline{CE} Controlled) [9, 14, 15]



Notes:

- 13. Address valid prior to or coincident with \overline{CE} transition LOW.
- 14. Data I/O is high impedance if $OE = V_{IH}$.
- 15. If \overline{CE} goes HIGH simultaneously with \overline{WE} HIGH, the output remains in a high-impedance state.

Switching Characteristics Over the Operating Range^[6]

Parameter	Description	CY62256-55		CY62256-70		Unit
		Min.	Max.	Min.	Max.	
Read Cycle						
t _{RC}	Read Cycle Time	55		70		ns
t _{AA}	Address to Data Valid		55		70	ns
t _{OHA}	Data Hold from Address Change	5		5		ns
t _{ACE}	\overline{CE} LOW to Data Valid		55		70	ns
t _{DOE}	\overline{OE} LOW to Data Valid		25		35	ns
t _{LZOE}	\overline{OE} LOW to Low-Z ^[7]	5		5		ns
t _{HZOE}	\overline{OE} HIGH to High-Z ^[7, 8]		20		25	ns
t _{LZCE}	\overline{CE} LOW to Low-Z ^[7]	5		5		ns
t _{HZCE}	\overline{CE} HIGH to High-Z ^[7, 8]		20		25	ns
t _{PU}	\overline{CE} LOW to Power-up	0		0		ns
t _{PD}	\overline{CE} HIGH to Power-down		55		70	ns
Write Cycle^[9, 10]						
t _{WC}	Write Cycle Time	55		70		ns
t _{SCE}	\overline{CE} LOW to Write End	45		60		ns
t _{AW}	Address Set-up to Write End	45		60		ns
t _{HA}	Address Hold from Write End	0		0		ns
t _{SA}	Address Set-up to Write Start	0		0		ns
t _{PWE}	\overline{WE} Pulse Width	40		50		ns
t _{SD}	Data Set-up to Write End	25		30		ns
t _{HD}	Data Hold from Write End	0		0		ns
t _{HZWE}	\overline{WE} LOW to High-Z ^[7, 8]		20		25	ns
t _{LZWE}	\overline{WE} HIGH to Low-Z ^[7]	5		5		ns