Atividade 23 - Direct Memory Access

Vinicius Gasparini

29 de agosto de 2020

1 DMA

Direct Memory Access (DMA) é um procedimento que algumas aplicações realizam a fim de encurtar o *pipeline* entre a memória e o disco ou outro tipo de dispositivo, evitando então o processador.

Nossos computadores operam, em suma maioria, sob dados de entrada transformados pelo processador e então é dado a saída. Porém alguns tipos de dados não necessitam dessa tal transformação que é realizada pelo processador. Por exemplo, uma placa de captura de vídeo pode não necessitar de trabalho da CPU do computador. Através do acesso DMA, o dispositivo de captura realiza leituras diretas na memória e realiza a interpretação desses dados no *hardware* da placa externa. Outro exemplo de uso são os discos DMA que com o auxilio de controladores IDE propiciam velocidades de transferência de dados sem necessidade de chamar o processador como intermédio.

Normalmente o único componente que acessa a memória RAM da máquina é o processador. Durante uma operação de *read* por exemplo, com DMA temos as seguintes etapas:

1. Quando o processador incia um *read*, um *buffer* é alocado e então são executadas algumas instruções que direcionam os dados para esse *buffer* enquanto o processo esta em congelamento.

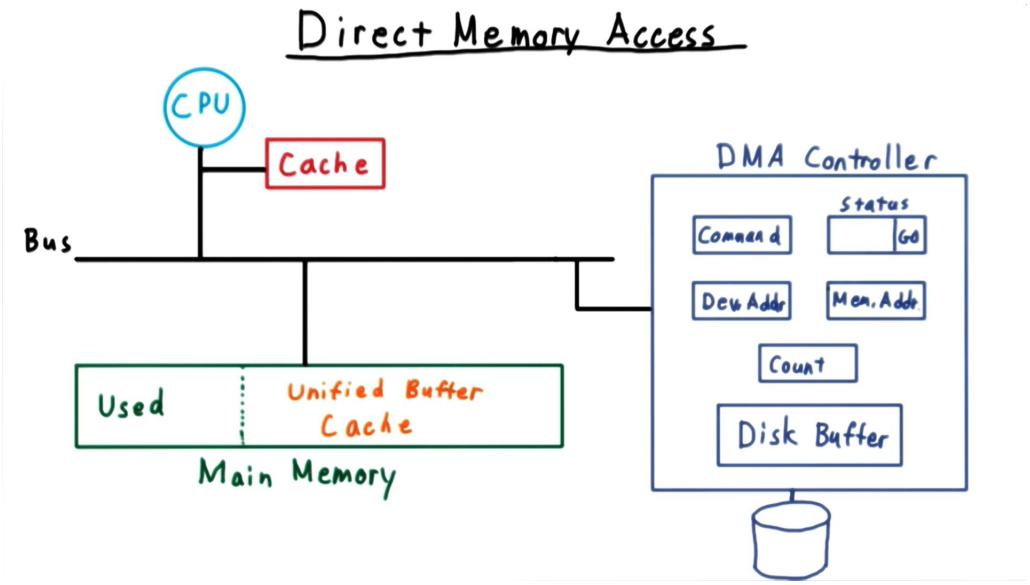
2. Finalizado a escrita no *buffer*, o processador saí do congelamento e da prosseguimento a leitura.

É possível que o DMA seja executado de maneira *assíncrona* com a alteração que o *pipeline* não fica estagnado enquanto os dados são lidos ao *buffer*.

De maneira simplificada então, DMA realiza tanto entrada de dados quanto saída através de blocos de informação acumuladas em um *buffer*. Esse *buffer*, externo ao circuito do processador, é independente e muito rápido. Todos os computadores modernos utilizam esse procedimento devido a evolução do volume de dados que utlizamos.

Porém, por ser um circuito extra ao *pipeline* padrão de dados, a alocação e manejo desse *buffer* carece de atenção. Então é definido um *DMA controller* que fará o manejo

de lançar exceções e interrupções de acordo com a necessidade, armazenará variáveis de controle de estado e será a ponte entre o disco/dispositivo de E/S.



Esquema simplificado do uso de DMA — Autor: Charles Brubaker

No esquema simplificado acima podemos ter uma noção básica do funcionamento do DMA. A CPU informa ao *controller* do dispositivo o tamanho da informação que será lida/gravada (*count*), o endereço do dispositivo (neste caso, a porta do disco), o endereço de memória do início da informação e a instrução de comando (leitura ou escrita). Sob posse de todos estes dados, o *controller* inicia o processo de deslocamento da informação e armazena no *buffer*. Finalizado essa etapa, é sinalizado a CPU e então é liberado o *pipeline* para execução de outras tarefas. Os dados já estão em *buffer* e agora só restam ser gravados no disco.

2 Referências

CHARLES BRUBAKER, *Advanced Operating Systems - GT Online Course*. Udacity, 2015.

CORBET J., RUBINI A., KROAH-HARTMAN G.*Linux Device Drivers*. Memory Mapping and DMA. LWN, 2005.