

Periferie
Oplossingen examenvragen

Tim Besard

14 juni 2010

Hoofdstuk 1

Grote vragen

Hoe lees je een bestand “test.txt” uit dat zich in de hoofddirectory van een FAT32 bestandssysteem bevindt (en leg hierbij de algemene structuur van het bestandssysteem uit, zoals de locatie en grootte van opgeslagen gegevens)? Hoe bereken je de startcluster van een bestand? Wat is VFAT?

De eerste stap hiertoe is het ophalen van de hoofddirectory. De startcluster daarvan is te vinden in de FAT32-bootsector, en na het overlopen van de FAT-tabel kan je de inhoud van de hoofddirectory ophalen.

Door die gegevens te interpreteren als een lijst van directory-entries, kan de directory overlopen worden. Eenmaal het clusternummer van het gezochte bestand gevonden is (door bij elke entry zowel het naamveld als extensieveld te vergelijken met de te zoeken informatie), moet de FAT tabel (die zich bevindt na de bootsector) daarmee geïndexeerd worden, waarna dit sequentieel moet voortgezet worden tot alle clusters van het bestand gevonden (en eventueel van de schijf opgehaald) zijn. Merk op dat bij het verwerken van de directory evenzeer de FAT tabel moet geraadpleegd worden om de relevante clusters op te halen.

VFAT tenslotte is een uitbreiding van FAT om langere bestandsnamen toe te laten. Bij FAT32 was dit echter gelimiteerd tot 8 karakters, en is ook als vast veld geëncodeerd binnen de structuur van een directory-cluster. Als oplossing, die tevens backwards-compatibel is, zal VFAT elk bestand met een lange bestandsnaam voorzien van een aantal extra entries binnen de directory, maar ze een ongeldige set aan attributen geven zodat oudere systemen de pseudo-bestanden kunnen negeren. Systemen die de attribuut-combinatie wel ondersteunen, weten dat ze die moeten interpreteren als pseudo-entry die op een langere bestandsnaam duidt.

Geef de verschillende componenten van de grafische kaart, en leg steeds het nut en de mogelijkheden ervan uit. Geef de mogelijke soorten videogeheugen. Wat is GPU Computing?

Een grafische kaart bestaat altijd uit de volgende componenten:

- **BIOS:** deze code, die zich bevindt in een BIOS ROM op de kaart, staat in voor de initialisatie van de hardware. Daar ze een `aa55` identificatie bytecode bevat, zal ze tijdens de POST geladen worden door de moederbord BIOS.
- **Processor:** dit is een massief-parallele variant van de CPU, die instaat voor de verwerking van grafische taken. Meestal gebeurt dit via een specifiek grafisch-georiënteerde API (zoals OpenGL of DirectX), maar general-purpose alternatieven worden recentelijk ook beschikbaar (CUDA, OpenCL, DirectCompute). Het gebruiken van de grafische APIs voor general-purpose doeleinden heet men “General Computing on GPU” terwijl het gebruik van een specifieke API (met bijhorende hardwareondersteuning) “GPU Computing” of “General Purpose GPU computing” doopt.
- **RAMDAC:** dit is een digitaal naar analoog converter uitgerust met een kleine hoeveelheid RAM, die instaat voor de omzetting van interne data naar de corresponderende signalen. Een grafische kaart bevat intern meestal 3 RAMDACs (voor elk kleur 1), en het is een essentiële component van de pipeline daar ze bepalend is voor de uiteindelijke verwerkingssnelheid van de kaart.
- **Driver:** dit is de besturingssysteem-specifieke schakel die instaat voor de verwerking van software API calls via uitvoering op de grafische kaart.
- **Geheugen:** waar low-cost grafische kaarten gebruik maken van het hoofdgeheugen op het moederbord (shared memory), zullen high-end kaarten uitgerust zijn met dedicated geheugen, meestal geoptimaliseerd naar snelheid toe.

Hoe worden gegevens op DVD en BluRay schijven beveiligd?

In het geval van de DVD zijn er verschillende technieken voorhanden:

- **Region Playback Control (RPC):** hierbij voorziet de DVD in een byte die indiceert waar de media mag afgespeeld worden. De DVD speler moet vervolgens controleren of die code wel overeenkomt met de huidige regiocode, en indien niet het afspelen weigeren. Sommige DVD-spelers kunnen echter alle regiocodes afspelen, en heet men dan ook “region free”. Computer-drivers gebruiken RPC-II, waarbij de regiocode van de drive (theoretisch) 4 keer mag veranderd worden, waarna de code vastgelegd wordt op de laatste ingestelde regiocode.
- **Content Scramble System (CSS):** deze technologie encrypteert de data op de DVD via verschillende sleutels: namelijk de title-key (die instaat voor de versleuteling van een bepaald hoofdstuk) en de disc-key (die instaat voor de versleuteling van de regio waar de title-keys zich bevinden). Elke DVD bevat een geëncrypteerde sectie met de disc-key in, die enkel kan ontsleuteld worden door gebruik te maken van de player-key, uniek aan elke DVD-speler. Het probleem hierbij is dat de regio met de disc-key in, verschillende keren moet voorkomen: namelijk geëncrypteerd voor elke mogelijke DVD player key (dat zijn er 400). Zo resulteert een enkele gecompromiteerde player-key in een manier om elke DVD volledig te decrypteren.

- **Analog Protection System (APS)**: hierbij voorziet de videodata in signalen die niet door een TV weergegeven worden, maar toch een analoge kopiëermechanisme in de war sturen. Dit werd vooral toegepast bij VHS banden.

Het recentere BluRay formaat kent gelijkaardige kopiëerbeveiligingen:

- **Region Playback Control (RPC)**: analoog aan het DVD systeem, maar beperkt tot 3 regiocodes.
- **Advanced Access Content System (AACS)**: deze technologie is vergelijkbaar met zijn tegenspeler bij DVD (CSS), maar zal gecompromitteerde player-keys niet langer voorzien van de mogelijkheid om de disc-key te decoden. Men kan dus selectief bepaalde BluRay spelers onklaar maken door hun unieke decryptiesleutels in te trekken.

Bespreek het vermogen van de processor, en leg het verschil uit tussen dynamisch en statisch vermogen. Bespreek tevens koeling bij een processor, waarmee moet rekening gehouden worden?

Het vermogen van een processor wordt gesplitst in twee componenten: het **dynamisch vermogen** dat afhankelijk is van het aantal transistor transitieën, en het statisch vermogen dat altijd aanwezig is. Het dynamisch vermogen wordt veroorzaakt door twee fenomenen. Eerst is er het **short circuit current**, dat plaatsvindt wanneer een schakelende CMOS-inverter (die instaat voor het inverteren van binnenkomende signalen), die theoretisch oneindig snel schakelt maar praktisch enige tijd nodig heeft om te schakelen, tijdens de schakelperiode de massa naar de positieve stroomlijn kortsluit. Het tweede fenomeen is de **switched capacitance**, dat veroorzaakt wordt door het vermogen benodigd om de vele elektronische schakelaars binnen de CPU van polariteit te wisselen. Dit vermogen valt uiteen in verschillende factoren, die elk apart kunnen toedragen aan een beperking van dit vermogen: Aantal transitieën verminderen: dit kan bekomen worden door inactieve delen van de processor tijdelijk los te koppelen van de klokfrequentie (clock gating). Capaciteit van de schakeling verminderen: door de grootte van de transistoren te verkleinen, slaan die minder lading op waardoor de capaciteit afneemt. Core spanning laten afnemen: aangezien dit een kwadratisch effect heeft wordt dit het meest toegepast. Klokfrequentie verlagen: dit wordt meestal samen toegepast met het afnemen van de spanning, aangezien bij een spanningsreductie de vertragingstijden moeten toenemen (vandaar de naam dynamic voltage scaling).

Het **statisch vermogen** tenslotte wordt veroorzaakt door de lekstromen die bij iedere transistor aanwezig zijn. Momenteel ligt dit vermogen nog veel lager dan het dynamische vermogen, maar door de recente toename van het aantal transistoren zal dit naar alle verwachting toch een belangrijke rol gaan spelen. Koeling is nodig om het vermogen dat processors genereren (bij recente CPU's soms tot 100 W), dat resulteert in een opwarming van het oppervlak van de processor, af te voeren naar een groter oppervlak waar het kan afkoelen zonder schade aan te richten.

Waar en waarom is encoding nodig? Geef enkele voorbeelden en bespreek.

Encodingen worden toegepast om gegevens om te zetten naar een formaat waar ze beter geschikt zijn voor het medium waar ze op opgeslagen worden. Dit wordt meestal toegepast om enerzijds het aantal bittransities te beperken, en anderzijds om te zorgen dat de transities in kwestie niet te ver van elkaar plaatsvinden om synchronisatieproblemen te vermijden. Een eerste voorbeeld zijn magnetische opslagmedia, waarbij een goede encoding zorgt voor een grotere capaciteit alsook een reductie van de kans op fouten. Daarbij worden de volgende encodingsmechanismen gebruikt:

- **Frequency Modulation (FM)**: een eenvoudige en verouderde techniek waarbij 1 of 2 transities gegenereerd worden naargelang de bit hoog of laag is.
- **Modified Frequency Modulation (MFM)**: deze verbetering van FM gebruikt nog steeds 2 bitcellen per databit, maar zal bij 0 bits nu enkel een transitie genereren in de eerste bitcel afhankelijk van het feit of de vorige bit ook laag was. Hierdoor kan 2 keer zoveel data geëncodeerd worden als bij FM.
- **Run Length Limited (RLL) x, y**: dit is geen concrete encodingsimplementatie: maar voorschrijft hoe het zich moet gedragen: minimaal x en maximaal y niet-transitionele bitcellen tussen twee opeenvolgende transities. Zo is FM RLL 0,1 en MFM RLL 1,3. Bij recente harde schijven gebruikt men RLL 1,7.

De techniek is ook terug te vinden bij optische media zoals CD's en DVD's: zo wordt bij CD's een eight to fourteen modulatie (EFM) gebruikt, wat in essentie een RLL 2,10 encoding is. Ook bij verschillende andere communicatievormen wordt encoding gebruikt, bijvoorbeeld bij SATA of PCIe (om een kloksignaal te moduleren in de datastream).

Bespreek 3 redenen waarom men de ATA-standaard zou aanpassen.

De ATA standaard is door de jaren heen verschillende keren aangepast, om verschillende redenen:

1. **Capaciteit**: verschillende keren bleek de bovengrens van hoeveelheid data onvoldoende, en heeft men de moeten aanpassen. Eerst door gebruik te maken van een virtuele CHS-notatie (omdat er een verschil is wat betreft CHS notatie tussen het BIOS en de ATA specificatie). Een tweede stap maakte het gebruik van LBA mogelijk, maar met slechts een 28 bit resolutie (BIOS limiet). De laatste stap maakt gebruik van Pure LBA, door de BIOS uit te breiden met de Enhanced Disk Drive (EDD) extensie. Hoewel deze extensie het gebruik van een 64-bit LBA voorschrijft, hebben de recentste ATA specificaties slechts voorzien in een 48-bit LBA.
2. **Snelheid**: initiëel maakte men enkel gebruik van Programmed IO (PIO), waardoor de overdrachtssnelheden sterk beperkt blijven. Sindsdien maakt men gebruik van

steeds krachtigere DMA mechanismen, zoals nu Ultra DMA dat zelfs een aparte DMA chip nodig heeft. Men heeft er ook het kabeiformaat voor moeten vernieuwen (om elektrostatische ruis te verminderen).

3. **Mogelijkheden:** om meer devices te ondersteunen heeft men de ATA standaard, eerst enkel gericht op harde schijven, ook vernieuwd: de ATAPI standaard (ATA Packet Interface).

Bespreek uitvoerig High Sierra en ISO9660. Wat is Joliet?

High Sierra is een bestandssysteem specifiek gemaakt voor optische media. Het is toendertijd gemaakt om de leesbaarheid van een CD-ROM op alle systemen mogelijk te maken, zonder steeds een gepaste driver te moeten ontwikkelen om een specifiek bestandssysteem te kunnen lezen. Het is ook geoptimaliseerd voor read-only media, door buiten de standaard FAT tabel (en diens recursieve zoekmethode) ook te voorzien en een specifieke path-tabel, waarmee opzoeken sneller kunnen gebeuren.

Het High Sierra formaat is uiteindelijk gestandaardiseerd onder **ISO 9660**, dat voorziet in 3 niveau's van ondersteuning. Het eerste niveau is het striktste (8.3 naamgeving, maximaal 8 niveaus diep, aansluitende bestanden), en wordt door bijna iedere computer ondersteund. Level 2 breidt dit uit door de maximale lengte van bestandsnamen op te drijven tot 30, en level 3 maakt het mogelijk om verschillende bestanden niet-aaneensluitend op te slaan.

Joliet tenslotte is een Microsoft-specifieke uitbreiding van ISO 9660 die de volgende verbeteringen toevoegt:

- Bestandsnamen kunnen tot 64 (Unicode) karakters bevatten
- Directories kunnen een extentie hebben en 8 niveaus diep gaan
- Multisessie

Bespreek de spoor- en sectorlayout bij een diskette en bij een harde schijf. Wat zijn de belangrijkste verschillen?

Beide spooropmaken voorzien in een strikte layout met verschillende synchronisatiepatronen alsook lege stukken die overlapping vermijden en de controller tijd geven voorgaande informatie te verwerken. Bij een harde schijf is het aantal synchronisatiepatronen en gaps aanzienlijk minder dan bij een floppy. Dit is te verklaren wegens het feit dat synchronisatie bij floppies veel belangrijker is daar de rotatiesnelheid sterk variabel is (vertraagt door het fysiek contact tussen de leeskop en de datadrager).

Meestal maakt men ook gebruik van **zoned-bit recording**: waarbij de schijf onderverdeeld wordt in zones. Het aantal sectoren per zone is daarbij constant, maar het aantal zones per spoor kan verschillen om de reductie aan schijfoppervlak aan de binnenkant van de schijf te compenseren. Hierdoor kan men profiteren van de grotere rotatiesnelheid aan de buitenkant van de schijf, en is er geen nood meer aan een variabele rotatiesnelheid.

Ook past men vaak **interleaving** toe, waarbij sectoren niet sequentieel plaatsvinden op de schijf maar met andere sectoren ertussen gevoegd (1 bij enkelvoudige interleaving, 2 bij dubbele interleaving). Dit geeft de controller tijd om de gelezen data te verwerken, zonder daarbij te moeten wachten op de volgende rotatie.

Tenslotte kent men nog **cylinder skewing**, waarbij het startpunt van verschillende cylinders enigszins verschilt om een overgang van de ene cylinder naar de andere (die enige tijd in beslag neemt) te compenseren zodat eens te meer niet moet gewacht worden op een hete rotatie. Hetzelfde principe wordt toegepast bij de overgang naar een andere leeskop, head skewing genaamd, waarbij het verschil echter veel kleiner is daar de overgang naar een andere kop slechts een elektrische operatie is.

Bespreek alle types RAM die afgeleid zijn van DRAM.

DRAM, dat gebruik maakt van condensatoren voor het bewaren van een toestand, is de basis geweest van verschillende geheugens. Zo is er het **Fast Paged Mode DRAM** (FPM DRAM), dat het geheugen voorstelt als een 2D matrix. Het voordeel hierbij is dat eenmaal een bepaalde rij geadresseerd is, men het rijnummer bij opeenvolgende fetches binnenin dezelfde rij niet moet herhalen. Een andere optimalisatie die hierbij toegepast werd is dat een enkele fetch volgens het lokaliteitsprincipe ook data van de drie volgende adressen doorstuurt. Een laatste optimalisatie, interleaving, zorgt ervoor dat aanliggende bytes verdeeld worden over verschillende geheugenbanken, zodat bij het versturen van de ene byte, de opeenvolgende byte reeds kan opgehaald worden om de delay tussen de twee bytes te minimaliseren.

Een volgende afgeleide, **Extended Data Out DRAM** (EDO DRAM), schrijft voor dat bij het zenden van data een nieuw adres reeds tegelijk mag verstuurd worden zodat aansluitend de data van het volgende adres reeds kan worden bewerkt.

Bij **Synchronised DRAM** (SDRAM) vervolgens wordt afgestapt van het asynchrone principe en zal men gegevensoverdracht toelaten op discrete tijdstippen. De klok is daarbij gesynchroniseerd met de systeemklok en bijgevolg met de processor.

Double Data Rate SDRAM (DDR DRAM) verdubbelt de snelheid van SDRAM zonder de kloksnelheid daarbij te doen stijgen, door op beide flanken van de klok datatransfers toe te laten. Double Data Rate 2 SDRAM (DDR2 SDRAM) breidt uit op dit concept door 2 geleiders te gebruiken per datalijn. Het differentiële karakter daarbij zorgt voor minder elektromagnetische ruis waardoor de kloksnelheid kan opgedreven worden.

Rambus DRAM (RDRAM) tenslotte maakt gebruik van narrow-channel transport (16-bit databus in plaats van 64-bit bij SDRAM), waardoor de kloksnelheid aanzienlijk kan opgedreven worden. Transfers vinden ook plaats op beide flanken.

Bespreek het moederbord en de verschillende I/O bussen.

Het moederbord bestaat uit verschillende componenten. Enerzijds is er het **BIOS**, de schakel tussen soft- en hardware. De BIOS kent verschillende functies, zo zal er een Power on self-test (POST) uitgevoerd worden, en kan de gebruiker bepaalde instellingen

wijzigen in het BIOS Setup Utility. Natuurlijk voorziet de BIOS in bepaalde routines die het besturingssysteem ondersteunen, alsook kan de BIOS gebruikt worden om hardware te monitoren.

Vervolgens hebben we de **processor**, die instaat voor de verwerking van instructies. Meestal bevindt die zich op een socket op het moederbord, maar bij oude architecturen (zoals de Pentium 2) werd voorzien in een dochterkaart die in een speciaal slot gestoken werd.

Om alle periferie te kunnen laten samenwerken is er voorzien in een **chipset**, waardoor er geen aparte controller meer nodig is voor elke periferiecomponent. Deze kan ontworpen zijn volgens de North & South Bridge architectuur, maar die wordt niet meer gebruikt omdat de PCI bus daarbij belast werd met irrelevante queries. Het alternatief is een Hub architectuur, waarbij de opdeling logischer is en snellere componenten dichterbij de processor zitten aangesloten op de Memory Controller Hub (MCH). Tragere periferie is aangesloten op de IO Controller Hub (ICH).

Natuurlijk bevat het moederbord ook verschillende **geheugenbanken**, die meestal maar 1 type geheugen ondersteunen. Tenslotte wordt ook voorzien in een batterij, die gebruikt wordt om de gebruikersinstellingen van de BIOS, opgeslaan in CMOS-RAM, bij te houden.

Wat betreft **I/O bussen** zijn er reeds verschillende systemen ontwikkeld. Zo was er eerst de **Industry Standard Architecture** (ISA) bus, een 16-bit bus die verschillende keren geüpgradet is met een hogere kloksnelheid.

Een verbetering op de ISA bus werd voorgesteld met de **Micro Channel Architecture** (MCA) bus, maar wegens hoge licentiekosten is deze nooit populair geworden. Als antwoord op MCA is de Extended ISA (EISA) bus geïntroduceerd, een 32-bit bus die voorzag in een tweede rij met aansluitcontacten (maar toch neerwaarts compatibel).

Aangezien het gebruik van een enkele I/O bus ervoor zorgt dat de bus nooit sneller kan zijn dan zijn traagste component, is soms voorzien in een speciale bus die zich dichterbij de processor bevindt: een lokale bus. Een voorbeeld van een dergelijke bus is de **Video Electronics Standards Association** (VESA) lokale bus. Door de drastische versnelling van processoren kon de periferie echter niet meer mee, waardoor lokale bussen in onbruik zijn geraakt.

Een volgende bus was de **Peripheral Component Interconnect** (PCI) bus, een moderne bus die populair is geworden voor het aansluiten van allerhande periferie.

De PCI bus was echter traag wat betreft geheugenoverdracht, waardoor men de **Accelerated Graphics Port** (AGP) bus heeft ontwikkeld. Deze bus, deels gebaseerd op de PCI-bus, zorgt voor een zeer snelle verbinding van de grafische kaart en het geheugen. Er zijn verschillende versies van ontwikkeld, die verschillen op gebied van voltage en kloksnelheid. Om mee te kunnen zou de architectuur echter moeten herzien worden, iets dat nooit gebeurd is.

De AGP bus is opgevolg door de **PCI-Express** (PCIe) bus, die net als zovele recente ontwikkelingen gebruik maakt van seriële communicatie. De bus werkt full-duplex, en om ruis te verminderen wordt gebruik gemaakt van differentiële geleiders. Door data te spreiden over verschillende lanes (interleaving) worden nog hogere snelheden bereikt.

Bij gebrek aan apart kloksignaal, wordt gebruik gemaakt van een 8B/10B encoding om het kloksignaal te encoderen in de datastroom.

Bespreek LVM.

In Linux systemen is het mogelijk de logische voorstelling van volumes uit handen te geven aan de Logical Volume Management (LVM) service, waarbij twee extra lagen geïntroduceerd worden tussen de partities en het effectieve bestandssysteem.

Eerst zijn er de **volume groups** (VG), samengesteld uit verschillende **physical volumes** (PV) wat praktisch partities of volledige schijven zijn. Zo kan men wat fysiek bestaat uit verschillende schijven of partities, samenvoegen tot een logisch geheel. Ook is het mogelijk dynamisch fysieke eenheden toe te voegen of te vermijden, waarbij de nodige data automatisch gemigreerd wordt.

Vervolgens zijn er de **logical volumes** (LV), logische eenheden die onderdeel zijn van een volume group, waarop een gebruiker een bestandssysteem kan aanmaken. Deze eenheden laten toe dat ze dynamisch vergroot of verkleind worden (wat natuurlijk enkel nuttig is als het onderliggende bestandssysteem dat ook ondersteund). Bij beide lagen zijn de eenheden steeds ondubbelzinnig te benoemen, wat het gebruiksgemak natuurlijk enkel ten goede komt. Ook kan eenvoudig een snapshot van de data gemaakt worden.

Bespreek SATA, SAS en PNP-SCSI.

Serial ATA (SATA) is een uitbreiding van het ATA protocol waarbij gebruik gemaakt wordt van een seriële protocol in plaats van de parallele aanpak die ATA kenmerkt. Hoewel hierbij het aantal datalijnen vermindert, blijkt de uiteindelijke snelheid hoger te liggen daar er veel minder elektromagnetische ruis plaatsvindt in het sterk gereduceerde aantal geleiders. De bekabeling is dan ook niet langer de standaard 80-pin kabel, maar een 7-pin kabel waarbij gebruikt wordt van differential NRZ codering. Er zijn voor het moment 3 SATA standaarden, die verschillen in snelheid alsook het aanbod commandos.

Serial Attached SCSI (SAS) is opnieuw een seriële omvorming van het klassiek parallele SCSI systeem. Hierbij is zowel gekeken naar het SATA protocol wat betreft economische voordelen, maar heeft men toch de robuustheid en betrouwbaarheid van de SCSI-interface behouden. De SAS interface is ook compatibel met SATA, en ondersteunt ook expanders waarmee het aantal apparaten sterk toe kan nemen.

Plug And Play SCSI (PNP-SCSI) tenslotte is een plug-and-play uitbreiding van de klassieke SCSI interface. Het plug-and-play concept zorgt er vooral voor dat de SCSI bus automatisch afgesloten wordt, maar kent toch volledige neerwaartste compatibiliteit met het klassieke SCSI. Een tweede facet van plug-and-play is dat een enkele driver zou moeten volstaan om een device te configureren, wat bij klassieke SCSI onmogelijk is daar de driver zowel besturingssysteem- als controllerafhankelijk is. Om dit op te lossen zijn uniforme APIs geïntroduceerd: de Advanced SCSI Programming Interface (ASPI) en de Common Access Method (CAM).

Bespreek de buck converter, en leg tevens het basisprincipe van een DC naar DC converter uit.

Na het aftransformeren en gelijkrichten van een ingangsspanning (vb. 220 V AC naar 12 V D) hebben we meestal een onstabiele voeding, die wel eens durft variëren in spanning (zeker in combinatie met een variabele belasting). Daarom moeten we de spanning stabiliseren, waarvoor er verschillende technieken bestaan.

Een **DC naar DC converter** doet dit door een ingangs DC signaal om te zetten naar een andere spanning, maar opnieuw een DC signaal. Er zijn verschillende manieren om dit te realiseren, en een **buck converter** probeert dit zonder veel vermogen te verkwisten (in tegenstelling tot een ruisloze maar veel verbruikende spanningsdeler). Daartoe voorziet een buck converter in een dynamisch geregeld PWM signaal dat bepaald of de ingangsspanning gebruikt wordt om een condensator op te vullen, of tijdelijk onderbroken is waarbij de condensator zijn energie gebruikt om een uitgangsspanning te voorzien. Om bij het ontladen eenzelfde spanning te krijgen, wordt de stroom afgevlakt via een tweede-orde integratieschakeling (wat jammer genoeg zorgt voor een hoogfrequente ruis, die wel nog kan verminderd worden door de uitgangsspanning aan een bypass te leggen). Naarmate de belasting stijgt of daalt, zal het circuit zichzelf regelen door het PWM signaal te wijzigen, waardoor een andere uitgangsspanning bekomen wordt.

Hoofdstuk 2

Kleine vragen

Wat zijn de verschillende soorten partities, en waar wordt de relevante data opgeslaan?

Elementair bestaan er 2 soorten partities. Eerst zijn er de **primaire partities**, die volledig gedefiniëerd zijn binnen de MBR van een schijf. Aangezien de MBR een vaste grootte kent, is er dus maar ruimte voor 4 primaire partities. Om deze beperking te omzeilen, moet 1 van die partities de rol vervullen van **uitgebreide partitie**. Een dergelijke partitie bevat aan het begin van zijn datagebied een regio waarbinnen subpartities gedefiniëerd kunnen worden, de extended MBR (EMBR). Dergelijke subpartities heet men **logische partities**, en kunnen naar believen aangemaakt worden. De EMBR bevindt zich net voor de logische partitie, en bevat twee ingangen: eerst de relatieve positie binnen het EBR en de grootte van de partitie, en vervolgens de beginsector van het vervolg van de EBR en het aantal vrije sectoren van de uitgebreide partitie.

Leg de termen MTTF, MTTR en MTBF uit. Hoe worden ze gebruikt om de beschikbaarheid van een systeem te berekenen?

Deze termen worden gebruikt bij het uitdrukken van de beschikbaarheid en het verwachte falen van een systeem. Ze staan voor **Mean Time To Failure** (MTTF), **Mean Time To Repair** (MTTR), en de som daarvan zijnde de **Mean Time Between Failure** (MTBF). De beschikbaarheid van het systeem kan berekend worden door de MTTF te delen door de som van de MTTF en de MTTR. Om de MTBF van een heel systeem te berekenen, moet de inverse som van de MTBF waarden van alle deelcomponenten berekend worden. Ook kan berekend worden hoe sterk redundantie de beschikbaarheid verbeterd, de MTTF van een redundant paar wordt berekend door de kans op valen ($1/\text{MTTF}$) te verdubbelen, maar te vermigvuldigen met de kans dat dat een falen valt binnen de tijd dat een andere component gefaald is (MTTR/MTTF). De uiteindelijke beschikbaarheid is de inverse van deze probabilistische berekening.

Wat is de maximumgrootte van een bestand en een partitie bij FAT32?

Een bestand kan maximaal 4 GB in beslag nemen, daar het grootte-veld van een bestandsingang in een directory maar 4 bytes in beslag neemt. Door het aantal ingangen in de FAT tabel te verhogen kan een partitie nu maximaal 2 TB in beslag nemen.

Bespreek software interrupts. Wat is er aan de in- en uitvoer nodig?

Software interrupts worden gebruikt om asynchroon met de processor te communiceren. Dit is uitermate handig wanneer de processor moet wachten op externe periferie: zo moet er geen gebruik gemaakt worden van busy waits, maar kan de uitvoering van andere code voortgezet worden totdat de periferie te kennen geeft dat de aanvraag verwerkt is.

Bespreek EFI, en het verschil met BIOS.

Extensible Firmware Interface (EFI) is een potentiële opvolger van het BIOS systeem, gemaakt voor de Itanium processorreeks. Het is bedoeld om het verouderde BIOS systeem te vervangen, en vervangt daarbij de MBR ook door het betere GUID Partition Table systeem (dat puur met LBA werkt, meer partities ondersteunt, en robuuster is). Ten behoeve van neerwaartse compatibiliteit bevindt zich echter een MBR op LBA 0.

Bespreek de verschillende groepen signalen bij PCI.

Eerst zijn er de **verplichte pinnen**:

- Adres- en datapinnen (AD, C/BE en PAR)
- Interface controle (FRAME, TRDY, IRDY, DEVSEL, STOP en ID SEL): pinnen die de transactie controleren.
- Error reporting (PERR en SERR): pariteitsfout of systeemfout.
- Arbitrage (REQ en GNT): om te bepalen welke initiator controle over de bus krijgt, meestal ingebouwd in de PCI chipset.
- Systeem (CLK en RST): het kloksignaal, en een asynchrone reset.

Vervolgens kan een toestel nog voorzien in bepaalde **optionele pinnen**:

- 64-bit uitbreiding (AD, C/BE, PAR64, REQ64, ACK64)
- Interface controle (LOCK): verzorgt atomaire toegang tot een target.
- Interrupts (INTA, INTB, INTC, INTD)
- Cache (SBO, SBDONE)
- JTAG (TDI, TDO, TCK, TMS, TRST)

Wat is Reflection Wave Switching?

Het nuttig gebruiken van een reflecterende golf (hiervoor zijn de afsluitweerstand en dus weggelaten). Dit wordt onder andere bij PCI toegepast: een initiële signaal zal dan maar half zo sterk zijn als benodigd, maar vervolledigd worden door zijn reflectie. Aangezien dit binnen de tijd van een klokcyclus moet gebeuren, is de lengte van de PCI bus dus beperkt.

Waarom is de A20 lijn geblokkeerd bij het opstarten van een PC? Wanneer wordt ze gedeblokkeerd?

Bij de x86 architectuur wordt meestal gebruik gemaakt van **gesegmenteerd geheugen**, waardoor een fysiek adres berekend wordt door een segmentadres te vermenigvuldigen met een bepaalde macht van 2, en het resultaat op te tellen bij het virtueel adres. Hierbij treedt een bepaalde overlap op, waardoor er meerdere mogelijkheden zijn om bepaalde bits te adresseren.

Gecombineerd met het feit dat de 8086 maar **1 MB** kon adresseren (wat overeenkomt met 20 adreslijnen, ofte A0 tot A20), bestaan er verschillende legale combinaties van een segment- en virtueel adres om het geheugenbereik net boven de 1 MB bovengrens aan te spreken. Hierbij wordt vervolgens **wrap around** toegepast, waardoor dergelijke adressen impact hebben op de laagste bits van het bereik. Die mogelijkheden zijn toegepast geweest om zonder het segmentadres te moeten aanpassen (wat traag was), verschillende delen van het geheugenbereik aan te spreken. Zo kon men, door het 16-bit segmentadres gelijk te stellen aan het maximum, de volgende hoeveelheid RAM boven de 1 MB grens adresseren: $2^{16} - 2^4$

Bij introductie van nieuwere adreslijnen (A20 en meer), werkten deze programma's natuurlijk niet meer. In plaats van deze te herschrijven, heeft IBM voorgeschreven dat een x86 processor initieel de A20 lijn moet uitschakelen, waardoor dergelijke programma's toch nog werken. Deze workaround bestaat nog altijd, waardoor bootloaders tijdens het opstarten expliciet de A20 lijn moeten inschakelen.