

2bb

May 18, 2024

Sheena Taylor 2b

Look at 2023 annual reports of Wal-Mart Download Wal-Mart and Costco Download Costco. The files are linked. Calculate and compare the cash-to-cash cycle/time for both companies. Submit a one-page report.

Title: Cash-to-Cash Cycle Analysis: Walmart vs. Costco

Introduction: The cash-to-cash cycle, also known as the cash conversion cycle, is a critical metric that measures the efficiency of a company's working capital management. It represents the time it takes for a company to convert its investments in inventory into cash from sales. A shorter cash-to-cash cycle is generally preferred as it indicates a more efficient use of working capital and a stronger cash flow position. In this report, we will analyze and compare the cash-to-cash cycles of two retail giants, Walmart Inc. and Costco Wholesale Corporation, based on their financial statements for the fiscal year 2023.

Methodology: The cash-to-cash cycle is calculated as follows:

Cash-to-Cash Cycle = Days Inventory Outstanding + Days Sales Outstanding - Days Payable Outstanding

Days Inventory Outstanding (DIO) represents the average number of days a company holds its inventory before selling it. Days Sales Outstanding (DSO) measures the average number of days it takes to collect cash from customers after a sale. Days Payable Outstanding (DPO) represents the average number of days a company takes to pay its suppliers.

Analysis:

Walmart Inc.:

Days Inventory Outstanding (DIO) = 50.60 days Days Sales Outstanding (DSO) = 5.52 days Days Payable Outstanding (DPO) = 48.00 days Cash-to-Cash Cycle = $DIO + DSO - DPO = 50.60 + 5.52 - 48.00 = 8.12$ days Costco Wholesale Corporation:

Days Inventory Outstanding (DIO) = 29.68 days Days Sales Outstanding (DSO) = Not reported (minimal receivables due to cash/debit sales model) Days Payable Outstanding (DPO) = 30.37 days Cash-to-Cash Cycle = $DIO - DPO = 29.68 - 30.37 = -0.69$ days Interpretation: Walmart's cash-to-cash cycle of 8.12 days indicates that it takes approximately 8 days for the company to convert its inventory investments into cash from sales, after accounting for the time it takes to collect from customers and pay suppliers. This relatively short cycle suggests that Walmart has efficient inventory management practices and a strong bargaining position with its suppliers, allowing it to delay payments while collecting cash from customers more quickly.

On the other hand, Costco's cash-to-cash cycle is negative at -0.69 days, which means that the company collects cash from sales before paying its suppliers. This unique situation is a result of Costco's business model, where it charges annual membership fees upfront and accepts only cash or debit card payments from members at its warehouses. Consequently, Costco has minimal accounts receivable and can rapidly convert inventory into cash, effectively being financed by its suppliers.

Costco's negative cash-to-cash cycle highlights its ability to minimize working capital requirements and finance its operations through supplier credit. This advantage allows Costco to maintain a lean cash conversion process and potentially invest the excess cash in other areas of the business.

Conclusion: While both Walmart and Costco exhibit efficient cash conversion processes, their cash-to-cash cycles differ significantly due to their distinct business models. Walmart's positive but relatively short cash-to-cash cycle of 8.12 days reflects its scale, bargaining power with suppliers, and effective inventory management practices. Costco's negative cash-to-cash cycle of -0.69 days is a unique advantage stemming from its membership-based model and cash/debit sales approach, enabling it to finance operations through supplier credit.

It is important to note that a negative cash-to-cash cycle is not inherently better or worse than a positive cycle; it depends on the company's specific industry, business model, and overall working capital management strategy. Both Walmart and Costco have demonstrated their ability to effectively manage their cash conversion cycles, contributing to their success in the highly competitive retail industry.

```
[ ]: # Walmart Data (in millions)
walmart_cost_of_sales = 443651
walmart_avg_inventory = 61281
walmart_revenue = 611329
walmart_avg_accounts_receivable = 9291
walmart_avg_accounts_payable = 58230

# Costco Data (in millions)
costco_merchandise_costs = 195952
costco_avg_inventory = 15662
costco_avg_accounts_payable = 18948

# Function to calculate days for a given metric
def calculate_days(numerator, denominator):
    if denominator == 0:
        return 0
    else:
        return 365 / (numerator / denominator)

# Walmart Calculations
walmart_days_inventory_outstanding = calculate_days(walmart_cost_of_sales, ↵
↵walmart_avg_inventory)
walmart_days_sales_outstanding = calculate_days(walmart_revenue, ↵
↵walmart_avg_accounts_receivable)
walmart_days_payable_outstanding = calculate_days(walmart_cost_of_sales, ↵
↵walmart_avg_accounts_payable)
```

```

walmart_cash_to_cash_cycle = walmart_days_inventory_outstanding +
    ↪walmart_days_sales_outstanding - walmart_days_payable_outstanding

# Costco Calculations
costco_days_inventory_outstanding = calculate_days(costco_merchandise_costs,
    ↪costco_avg_inventory)
costco_days_payable_outstanding = calculate_days(costco_merchandise_costs,
    ↪costco_avg_accounts_payable)
costco_cash_to_cash_cycle = costco_days_inventory_outstanding -
    ↪costco_days_payable_outstanding

# Print Results
print("Walmart Cash-to-Cash Cycle:")
print(f"Days Inventory Outstanding: {walmart_days_inventory_outstanding:.2f}
    ↪days")
print(f"Days Sales Outstanding: {walmart_days_sales_outstanding:.2f} days")
print(f"Days Payable Outstanding: {walmart_days_payable_outstanding:.2f} days")
print(f"Cash-to-Cash Cycle: {walmart_cash_to_cash_cycle:.2f} days")
print("\nCostco Cash-to-Cash Cycle:")
print(f"Days Inventory Outstanding: {costco_days_inventory_outstanding:.2f}
    ↪days")
print(f"Days Payable Outstanding: {costco_days_payable_outstanding:.2f} days")
print(f"Cash-to-Cash Cycle: {costco_cash_to_cash_cycle:.2f} days")

```

Walmart Cash-to-Cash Cycle:
 Days Inventory Outstanding: 50.42 days
 Days Sales Outstanding: 5.55 days
 Days Payable Outstanding: 47.91 days
 Cash-to-Cash Cycle: 8.06 days

Costco Cash-to-Cash Cycle:
 Days Inventory Outstanding: 29.17 days
 Days Payable Outstanding: 35.29 days
 Cash-to-Cash Cycle: -6.12 days

```

[ ]: # Walmart Data (in millions)
walmart_cost_of_sales = 443651
walmart_avg_inventory = 61281 # Calculated as (56576 + 56511) / 2 from balance
    ↪sheet
walmart_revenue = 611329
walmart_avg_accounts_receivable = 9291 # From balance sheet
walmart_avg_accounts_payable = 58230 # Calculated as (53742 + 55261) / 2 from
    ↪balance sheet

# Function to calculate days for a given metric
def calculate_days(numerator, denominator):

```

```

    if denominator == 0:
        return 0
    else:
        return 365 / (numerator / denominator)

# Walmart Calculations
walmart_days_inventory_outstanding = calculate_days(walmart_cost_of_sales,
    ↪walmart_avg_inventory)
walmart_days_sales_outstanding = calculate_days(walmart_revenue,
    ↪walmart_avg_accounts_receivable)
walmart_days_payable_outstanding = calculate_days(walmart_cost_of_sales,
    ↪walmart_avg_accounts_payable)
walmart_cash_to_cash_cycle = walmart_days_inventory_outstanding +
    ↪walmart_days_sales_outstanding - walmart_days_payable_outstanding

# Print Walmart Results
print("Walmart Cash-to-Cash Cycle:")
print(f"Days Inventory Outstanding: {walmart_days_inventory_outstanding:.2f}
    ↪days")
print(f"Days Sales Outstanding: {walmart_days_sales_outstanding:.2f} days")
print(f"Days Payable Outstanding: {walmart_days_payable_outstanding:.2f} days")
print(f"Cash-to-Cash Cycle: {walmart_cash_to_cash_cycle:.2f} days")

```

Walmart Cash-to-Cash Cycle:
 Days Inventory Outstanding: 50.42 days
 Days Sales Outstanding: 5.55 days
 Days Payable Outstanding: 47.91 days
 Cash-to-Cash Cycle: 8.06 days

```

[ ]: # Costco Data (in millions)
costco_merchandise_costs = 212586
costco_avg_inventory = 17279 # Calculated as (16651 + 17907) / 2 from balance
    ↪sheet
costco_avg_accounts_payable = 17665.5 # Calculated as (17483 + 17848) / 2 from
    ↪balance sheet

# Function to calculate days for a given metric
def calculate_days(numerator, denominator):
    if denominator == 0:
        return 0
    else:
        return 365 / (numerator / denominator)

# Costco Calculations
costco_days_inventory_outstanding = calculate_days(costco_merchandise_costs,
    ↪costco_avg_inventory)

```

```

costco_days_payable_outstanding = calculate_days(costco_merchandise_costs,
↪costco_avg_accounts_payable)
costco_cash_to_cash_cycle = costco_days_inventory_outstanding -
↪costco_days_payable_outstanding

# Print Costco Results
print("Costco Cash-to-Cash Cycle:")
print(f"Days Inventory Outstanding: {costco_days_inventory_outstanding:.2f}
↪days")
print(f"Days Payable Outstanding: {costco_days_payable_outstanding:.2f} days")
print(f"Cash-to-Cash Cycle: {costco_cash_to_cash_cycle:.2f} days")

```

Costco Cash-to-Cash Cycle:
 Days Inventory Outstanding: 29.67 days
 Days Payable Outstanding: 30.33 days
 Cash-to-Cash Cycle: -0.66 days